



Titre: Estimation de la direction d'arrivée d'un faisceau sur un réseau d'antennes en présence d'une réflexion parasite à l'aide de réseaux de neurones MLP
Title:

Auteur: Eric Danneville
Author:

Date: 2005

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Danneville, E. (2005). Estimation de la direction d'arrivée d'un faisceau sur un réseau d'antennes en présence d'une réflexion parasite à l'aide de réseaux de neurones MLP [Mémoire de maîtrise, École Polytechnique de Montréal].
Citation: PolyPublie. <https://publications.polymtl.ca/7604/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7604/>
PolyPublie URL:

Directeurs de recherche:
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

ESTIMATION DE LA DIRECTION D'ARRIVÉE D'UN FAISCEAU SUR UN
RÉSEAU D'ANTENNES EN PRÉSENCE D'UNE RÉFLEXION PARASITE À
L'AIDE DE RÉSEAUX DE NEURONES MLP

ERIC DANNEVILLE
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)
JUILLET 2005

© Eric Danneville, 2005.



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-16773-1

Our file Notre référence

ISBN: 978-0-494-16773-1

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

ESTIMATION DE LA DIRECTION D'ARRIVÉE D'UN FAISCEAU SUR UN
RÉSEAU D'ANTENNES EN PRÉSENCE D'UNE RÉFLEXION PARASITE À
L'AIDE DE RÉSEAUX DE NEURONES MLP

présenté par: DANNEVILLE Eric

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. BOSISIO Renato G., Ph.D., président

M. LAURIN Jean-Jacques, Ph.D., membre et directeur de recherche

M. BRAULT Jean-Jules, Ph.D., membre et codirecteur de recherche

M. NERGUIZIAN Chahé, Ph.D., membre

À mes parents, Alain et Annie

À ma famille et mes amis

REMERCIEMENTS

Je tiens à remercier tout d'abord mon directeur de recherche, le professeur Jean-Jacques Laurin, pour sa disponibilité, ses qualités humaines et ses conseils pertinents, très utiles, toujours saupoudrés de bonne humeur. Je le remercie également pour le financement qu'il a bien voulu m'accorder tout au long de ce projet, me permettant de le mener à terme dans de très bonnes conditions.

Je remercie aussi bien sûr mon co-directeur de recherche, le professeur Jean-Jules Brault, pour sa disponibilité à des moments parfois peu opportuns, son éclairage complémentaire et original sur le projet, et ses nombreuses anecdotes intéressantes.

Je remercie ensuite Steve Dubé et Roch Brassard pour leur soutien technique, leur savoir-faire et leur patience dans la mise en œuvre de ce projet. Mes remerciements s'adressent aussi à Jules Gauthier et Eric Marsan pour leur aide supplémentaire.

Enfin, je remercie ma famille et mes amis pour leur soutien, direct ou non, dans la vie de ce travail.

RÉSUMÉ

Ce mémoire porte sur un système de détection d'arrivée d'un faisceau par rapport à un réseau d'antennes disposées sur deux axes, X et Y , et à l'aide de réseaux de neurones de type MLP (ou "Multi Layer Perceptron") dans un cône d'ouverture 90° . Ils permettent d'estimer deux angles référés par rapport à X et Y , θ_x et θ_y respectivement, et de donner sans ambiguïté la direction d'arrivée (ou DOA en anglais) du signal dans un espace à trois dimensions. Ce système est également robuste au bruit comme l'avaient montré Michael Coudyser et Nelson Fonseca.

Une possible application de ce système de DOA serait la poursuite automobile. Or, dans ce cas, la route génère une réflexion parasite qui va perturber la qualité de la mesure. La première étape du travail a donc consisté à modéliser simplement mais néanmoins efficacement, une réflexion parasite, d'un point de vue électromagnétique. Celle-ci génère des interférences avec le faisceau principal (le signal direct). On a ensuite modifié les signaux de sorties des antennes ainsi que ceux utilisés par les réseaux de neurones pour tenir compte de cette réflexion parasite. Il a alors été possible de quantifier la perturbation induite sur l'estimation en terme d'erreur angulaire moyenne. On a pu dès lors comparer les performances de différentes modifications touchant aussi bien les antennes que les réseaux de neurones. Notre attention s'est particulièrement portée sur ces critères :

- choix de la polarisation des antennes,
- nombre d'antennes par réseau,
- nombre de MLPs et taille des signaux à leur entrée,
- caractéristique des MLPs en terme de nombres de neurones et d'époques d'entraînement,
- choix des points d'entraînement.

On a ainsi pu déterminer le système optimal de DOA en présence d'une réflexion parasite. Les résultats obtenus en simulation ont ensuite été confirmés par des

mesures prises sur un réseau d'antennes réel, d'axe Y . Elles ont été effectuées en chambre anéchoïde dans laquelle on avait placé un morceau d'asphalte pour modéliser l'effet d'une route.

Une erreur angulaire moyenne comprise entre 2 et 2.5° sur θ_y est obtenue aussi bien en simulations qu'expérimentalement en présence d'une réflexion parasite.

ABSTRACT

This thesis deals with a DOA system (for “Direction Of Arrival”) composed of a two axes (X and Y) antenna array, and using MLP (“Multi Layer Perceptron”) neural networks. The system operates in a cone whose vertex angle is 90° . They can estimate two angles, θ_x and θ_y , with respect to X and Y respectively. Thus, they provide DOA of an incident wave in a three dimensions space without ambiguity. The system is also robust to noise.

Car tracking could become a possible application for this DOA system. In this case, the road generates a parasitic reflection which perturbs the quality of the DOA estimation. The first step of this work consisted in simply albeit effectively modelling a parasitic reflection from an electromagnetic point of view. This reflected signal creates interference with the main beam (the direct signal). Antenna output signals and the ones feeding the neural networks are then modified to take this parasitic reflection into account. By this way, perturbation affecting the DOA estimation can be quantified in terms of average angular error. Performances of different modifications on the antennas or the neural networks have been compared. We focused on these main criteria :

- choice of antenna polarization,
- antenna number per array,
- MLPs number and size of feeding signals,
- MLPs characteristics (number of neurones and training epochs),
- choice of training points.

Thus, we succeed in optimizing the DOA system in presence of parasitic reflection. Then, simulation results have been confirmed by measurements on a real Y axis antenna array. They were carried out in an anechoic chamber in which a slab of asphalt was placed to model road effects.

The average angular error on θ_y varies between 2 and 2.5° in presence of reflection,

in both simulations and experiments.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES FIGURES	xiv
LISTE DES NOTATIONS ET DES SYMBOLES	xxi
LISTE DES TABLEAUX	xxvi
CHAPITRE 1 INTRODUCTION	1
1.1 Problématique générale	1
1.2 Revue de littérature	3
1.3 Le problème des réflexions parasites	5
1.4 Organisation du mémoire	7
CHAPITRE 2 SYNTHÈSE DES TRAVAUX PRÉCÉDENTS ET APPORTS	
PERSONNELS	9
2.1 Description générale du système	9
2.2 Structure du réseau d'antennes	12
2.2.1 Deux réseaux de trois antennes orthogonaux	12
2.2.2 Modélisation du faisceau d'arrivée et équations de sortie des antennes	13
2.3 Choix du combinateur de signaux	16

2.3.1	Choix des signaux théoriques	16
2.3.2	Optimisation des signaux réels	18
2.3.3	Utilisation des huit signaux de sortie du combinateur	22
2.3.4	Définition d'un seuil de perception pour l'estimation de la DOA	24
2.3.5	Normalisation et numérisation des vecteurs d'entrée des RNAs	24
2.4	Description et fonctionnement des MLPs utilisés	26
2.4.1	Qu'est-ce qu'un neurone artificiel?	26
2.4.2	Regroupement des neurones en réseaux	27
2.4.3	Principe de fonctionnement d'un MLP	29
2.5	Améliorations proposées par rapport aux travaux de Coudyser	32
2.5.1	Choix des points d'entraînement	32
2.5.1.1	Maillage arithmétique en θ et ϕ	33
2.5.1.2	Maillage optimal	36
2.5.2	Entraînement de MLPx (ou MLPy)	38
2.5.3	Entraînement de MLPd	40
2.5.4	Ajout du bruit et traitements pour en réduire son effet	42
2.5.4.1	Ajout du bruit de RSB fixé	44
2.5.4.2	Atténuation de l'effet du bruit sur l'estimation de la DOA par pré- et post-traitement, selon (Coudyser, 2003)	46
2.5.5	Résultats d'estimation avec du bruit et sans réflexion parasite avec MLPx et MLPy	47
CHAPITRE 3 EFFETS D'UNE RÉFLEXION PARASITE ET OPTIMI- SATION DE LA STRUCTURE DU SYSTÈME		55
3.1	Modélisation d'une route et de la réflexion par l'optique géométrique	55
3.2	Etude de la réflexion d'un point de vue électromagnétique	58

3.2.1	Polarisation d'une antenne	58
3.2.2	Coefficients de réflexion de Fresnel	60
3.2.3	Détermination expérimentale de coefficients de réflexion pour différents matériaux	63
3.2.4	Génération des signaux du système en présence de la réflexion parasite	65
3.3	Simulation de la robustesse du système à une réflexion parasite et au bruit	68
3.4	Intérêt du découplage de θ_x et θ_y	74
3.5	Etude de la polarisation des antennes	76
3.6	Comparaisons avec un réseau plus simple à deux paires d'antennes	79
3.7	Performances de MLPd	83
3.7.1	Présentation de l'étage de détection	83
3.7.2	Optimisations de MLPd	87
3.7.3	Figures des erreurs de classification de MLPd optimal	90
3.8	Conclusion	92
CHAPITRE 4 CHOIX DES POINTS D'ENTRAÎNEMENT DE MLPX ET MLPY		
4.1	Entraînement "sans réflexion" (D)	94
4.2	Entraînement "avec réflexion mono-configuration" (DR)	100
4.3	Entraînement "avec réflexion multi-configuration" (DRM)	103
4.4	Entraînement "bi-signal classique" (SS)	107
4.5	Entraînement "bi-signal décalé" (SSD)	112
CHAPITRE 5 VALIDATIONS EXPÉRIMENTALES		
5.1	Synthèse sous HFSS d'un système à trois antennes en guide d'onde	123
5.1.1	Choix et configuration des guides d'onde	123
5.1.2	Tracé des guides d'onde	125

5.1.3	Calcul de la taille du cornet	127
5.2	Description du dispositif expérimental	128
5.2.1	Diagramme de rayonnement des antennes	129
5.2.2	Coefficients de réflexion expérimentaux	130
5.2.3	Plans réflecteurs utilisés	131
5.3	Résultats expérimentaux sur l'estimation de θ_y dans le plan $\theta_x = 90^\circ$	133
5.3.1	Avec entraînement sans réflexion	136
5.3.2	Avec entraînement avec réflexion multi-configuration	139
5.3.3	Avec entraînement bi-signal décalé	141
5.3.4	Analyse des résultats expérimentaux	144
5.4	Résultats expérimentaux sur l'estimation de θ_y sur une portion d'es-	
	pace	145
5.4.1	Avec l'entraînement sans réflexion	146
5.4.2	Avec l'entraînement bi-signal décalé	148
5.4.3	Analyse des résultats	149
	CONCLUSION	151
	RÉFÉRENCES	154

LISTE DES FIGURES

FIG. 1.1	<i>Problématique de la poursuite automobile</i>	6
FIG. 2.1	<i>Schéma du système global d'estimation de la DOA</i>	10
FIG. 2.2	<i>Paire de réseaux d'antennes orthogonaux et définition des angles θ_x et θ_y</i>	12
FIG. 2.3	<i>Distance séparant les 3 antennes et déphasages associés . .</i>	13
FIG. 2.4	<i>Définition des angles du problème</i>	15
FIG. 2.5	<i>Signaux théoriques de présence et de précision en fonction de θ_x ou θ_y</i>	17
FIG. 2.6	<i>Sectorisation du cône d'étude en 16 parties, représentée à l'aide d'une portion de sphère unité en vue de dessus</i>	18
FIG. 2.7	<i>Formulation théorique d'un diviseur de puissance (1) et d'un coupleur hybride 90° (2)</i>	19
FIG. 2.8	<i>Formulation théorique du coupleur hybride 90° réellement utilisé dans le combineur</i>	20
FIG. 2.9	<i>Schéma du combineur de signaux de puissance optimal . .</i>	21
FIG. 2.10	<i>Signaux de sortie du combineur et signaux théoriques désirés en fonction de θ_x ou θ_y</i>	22
FIG. 2.11	<i>Signaux complémentaires du combineur en fonction de θ_x ou θ_y</i>	23
FIG. 2.12	<i>Signaux normalisés en fonction de θ_x ou θ_y</i>	25
FIG. 2.13	<i>Modèle de neurone artificiel</i>	26
FIG. 2.14	<i>Schéma d'un MLP classique à une couche cachée</i>	28
FIG. 2.15	<i>Fonctions de transfert rencontrées : en sigmoïde, linéaire et en seuil</i>	29
FIG. 2.16	<i>Maillage arithmétique en θ et ϕ</i>	34
FIG. 2.17	<i>Maillage arithmétique simplifié</i>	35

FIG. 2.18	<i>Maillage carré d'un disque de rayon unité à gauche, compression du maillage du disque à droite</i>	36
FIG. 2.19	<i>Maillage optimal des DOAs sur une calotte sphérique de rayon unité</i>	38
FIG. 2.20	<i>Maillage optimal pour entraîner MLPd</i>	41
FIG. 2.21	<i>Intérêt du bruit dans la phase d'entraînement d'un problème de classification</i>	43
FIG. 2.22	<i>Comparaison des signaux bruités avec le signal originel . . .</i>	45
FIG. 2.23	<i>Optimisation du nombre de neurones dans la couche cachée (sur 30 époques) à gauche, et du nombre d'époques d'entraînement (avec 30 neurones) à droite, pour MLPx et MLPy</i>	50
FIG. 2.24	<i>Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite</i>	51
FIG. 2.25	<i>Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx</i>	52
FIG. 2.26	<i>Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy</i>	52
FIG. 3.1	<i>Modèle de réflexion parasite utilisé dans nos simulations . .</i>	56
FIG. 3.2	<i>Coefficients de réflexion du sol en fonction de $\psi' = 90^\circ - \psi$ ($\epsilon_r = 15$, $\sigma = 12 \text{ e-}3$, $f = 77 \text{ GHz}$)</i>	61
FIG. 3.3	<i>C_h et C_v en fonction de ψ' et ϵ_r</i>	62
FIG. 3.4	<i>C_v théorique et expérimentale pour l'asphalte en fonction de ψ'</i>	64
FIG. 3.5	<i>C_v expérimentale pour divers matériaux en fonction de ψ'</i>	64
FIG. 3.6	<i>Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite, avec une réflexion parasite</i>	70
FIG. 3.7	<i>Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx avec une réflexion parasite</i>	70

FIG. 3.8	<i>Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec une réflexion parasite</i>	71
FIG. 3.9	<i>Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite, avec une réflexion parasite et 30 échantillons de pré-traitement</i>	71
FIG. 3.10	<i>Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx avec une réflexion parasite et 30 échantillons de pré-traitement</i>	72
FIG. 3.11	<i>Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec une réflexion parasite et 30 échantillons de pré-traitement</i>	72
FIG. 3.12	<i>Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite, avec une réflexion parasite et 30 échantillons de pré-traitement</i>	76
FIG. 3.13	<i>Coefficients de réflexion intervenant dans des polarisations horizontale, verticale et circulaire</i>	79
FIG. 3.14	<i>Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite, avec un réseau à deux paires d'antennes</i>	81
FIG. 3.15	<i>Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx avec un réseau à deux paires d'antennes</i>	81
FIG. 3.16	<i>Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec un réseau à deux paires d'antennes</i>	82
FIG. 3.17	<i>Etude de la complexité du classificateur en étudiant la sortie de MLPd en fonction des signaux de puissance du combineur, deux à deux (axe X)</i>	84
FIG. 3.18	<i>Etude de la complexité du classificateur en étudiant la sortie de MLPd en fonction des signaux de puissance du combineur, deux à deux (axe Y)</i>	85

FIG. 3.19	<i>Erreurs de classification de MLPd sous un RSB = 20 dB, sans réflexion</i>	90
FIG. 3.20	<i>Erreurs de classification de MLPd sous un RSB = 35 dB, avec réflexion, $\epsilon_r = 15$</i>	91
FIG. 4.1	<i>Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite avec entraînement D et $\epsilon_r = 3.2$</i>	97
FIG. 4.2	<i>Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx avec entraînement D et $\epsilon_r = 3.2$</i>	98
FIG. 4.3	<i>Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec entraînement D et $\epsilon_r = 3.2$</i>	98
FIG. 4.4	<i>Sorties moyennées et écart-type des estimations de θ_y par MLPy avec entraînement D et $\epsilon_r = 3.2$</i>	99
FIG. 4.5	<i>Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec entraînement D et $\epsilon_r = 15$</i>	99
FIG. 4.6	<i>Sorties moyennées et écart-type des estimations de θ_y par MLPy avec entraînement D et $\epsilon_r = 15$</i>	100
FIG. 4.7	<i>Limites de deux cônes d'intérêt différents avec le plan $\theta_x = 90^\circ$</i>	104
FIG. 4.8	<i>Fonction de densité de probabilité de d</i>	105
FIG. 4.9	<i>Comparaison des performances de l'entraînement D et de l'entraînement DRM, en fonction de la distance d</i>	106
FIG. 4.10	<i>DOAs du signal direct</i>	108
FIG. 4.11	<i>DOAs du signal réfléchi</i>	108
FIG. 4.12	<i>Exemple de décalage des DOAs à la base de l'entraînement SSD</i>	113
FIG. 4.13	<i>Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite avec entraînement SSD</i>	116
FIG. 4.14	<i>Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx avec entraînement SSD</i>	117

FIG. 4.15	<i>Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec entraînement SSD et $\epsilon_r = 3.2$</i>	117
FIG. 4.16	<i>Sorties moyennées et écart-type des estimations de θ_y par MLPy avec entraînement SSD et $\epsilon_r = 3.2$</i>	118
FIG. 4.17	<i>Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec entraînement SSD et $\epsilon_r = 15$</i>	118
FIG. 4.18	<i>Sorties moyennées et écart-type des estimations de θ_y par MLPy avec entraînement SSD et $\epsilon_r = 15$</i>	119
FIG. 4.19	<i>Comparaison des performances de l'entraînement D et de l'entraînement SSD, en fonction de la distance d</i>	120
FIG. 4.20	<i>Comparaison des performances de l'entraînement D et de l'entraînement SSD, en fonction de C_0</i>	121
FIG. 5.1	<i>Définition des longueurs a et b sur une section de guide d'onde</i>	124
FIG. 5.2	<i>Deux types de configuration possibles pour les guides d'onde : plan E ou plan H</i>	125
FIG. 5.3	<i>Tracé des guides d'onde du réseau d'antenne d'axe Y</i>	126
FIG. 5.4	<i>Coupe d'un guide et du cornet plan H dans le plan (X,Z)</i>	127
FIG. 5.5	<i>Photo des trois entrées des antennes en guide d'onde</i>	128
FIG. 5.6	<i>Diagramme de rayonnement des trois antennes, avec et sans cornet sectoriel plan H</i>	129
FIG. 5.7	<i>Coefficients de réflexion expérimentaux en entrée de nos trois antennes</i>	130
FIG. 5.8	<i>Photo du système monté sur le plan réflecteur en asphalte, dans la chambre anéchoïde</i>	132
FIG. 5.9	<i>Résultats d'estimation de MLPy (entraînement D) sans plan réflecteur, sans bruit additif, avec signaux $A_{yi,réel}$</i>	135
FIG. 5.10	<i>Résultats d'estimation de MLPy (entraînement D) sans plan réflecteur, sans bruit additif, avec signaux $A_{yi,repasé}$</i>	136

FIG. 5.11	<i>Résultats d'estimation de MLPy (entraînement D) testé avec un plan réflecteur en PVC</i>	137
FIG. 5.12	<i>Résultats d'estimation de MLPy (entraînement D) testé avec un plan réflecteur en contre-plaqué</i>	137
FIG. 5.13	<i>Résultats d'estimation de MLPy (entraînement D) testé avec un plan réflecteur en bitume caoutchouteux</i>	138
FIG. 5.14	<i>Résultats d'estimation de MLPy (entraînement D) testé avec un plan réflecteur en asphalte</i>	138
FIG. 5.15	<i>Résultats d'estimation de MLPy (entraînement DRM) testé avec un plan réflecteur en PVC</i>	139
FIG. 5.16	<i>Résultats d'estimation de MLPy (entraînement DRM) testé avec un plan réflecteur en contre-plaqué</i>	140
FIG. 5.17	<i>Résultats d'estimation de MLPy (entraînement DRM) testé avec un plan réflecteur en bitume caoutchouteux</i>	140
FIG. 5.18	<i>Résultats d'estimation de MLPy (entraînement DRM) testé avec un plan réflecteur en asphalte</i>	141
FIG. 5.19	<i>Résultats d'estimation de MLPy (entraînement SSD) testé avec un plan réflecteur en PVC</i>	142
FIG. 5.20	<i>Résultats d'estimation de MLPy (entraînement SSD) testé avec un plan réflecteur en contre-plaqué</i>	142
FIG. 5.21	<i>Résultats d'estimation de MLPy (entraînement SSD) testé avec un plan réflecteur en bitume caoutchouteux</i>	143
FIG. 5.22	<i>Résultats d'estimation de MLPy (entraînement SSD) testé avec un plan réflecteur en asphalte</i>	143
FIG. 5.23	<i>Résultats d'estimation de MLPy testé avec un plan réflecteur en asphalte, avec l'entraînement D (mesures en azimut et élévation)</i>	146

FIG. 5.24	<i>Résultats d'estimation en 3D de MLPy testé avec un plan réflecteur en asphalte, avec l'entraînement D (mesures en azimuth et élévation)</i>	147
FIG. 5.25	<i>Résultats d'estimation de MLPy testé avec un plan réflecteur en asphalte, avec l'entraînement SSD (mesures en azimuth et élévation)</i>	148
FIG. 5.26	<i>Résultats d'estimation en 3D de MLPy testé avec un plan réflecteur en asphalte, avec l'entraînement SSD (mesures en azimuth et élévation)</i>	149

LISTE DES NOTATIONS ET DES SYMBOLES

A_{xi}	Signaux de sortie de l'antenne i de l'axe X
$A_{xi,eb}$	Signaux de sortie de l'antenne i de l'axe X par l'entraînement bisignale
$A_{xi,r}$	Signaux de sortie de l'antenne i de l'axe X perturbés par une réflexion
$A_{xi,r}^h$	Signaux de sortie de l'antenne i à polarisation horizontale, perturbés par une réflexion
$A_{xi,r}^v$	Signaux de sortie de l'antenne i à polarisation verticale, perturbés par une réflexion
α	Constante complexe inconnue pondérant \vec{E}
\vec{B}	Vecteur champ magnétique
c	Vitesse de la lumière
C_h	Coefficient de réflexion de Fresnel horizontal
C_v	Coefficient de réflexion de Fresnel vertical
d	Distance entre l'émetteur et le récepteur
d_1	Distance séparant les antennes 1 et 2
d_2	Distance séparant les antennes 1 et 3
d_r	Distance parcourue par le rayon réfléchi
DOA^d	DOA du signal direct
DOA^r	DOA de la réflexion
$e_{3D}(j)$	Erreur d'estimation pour la j^{ieme} DOA
E	Emetteur
E_h	Composante horizontale de \vec{E}
E_t	Erreur d'entraînement
E_v	Erreur de validation

E_v	Composante verticale de \vec{E}
\vec{E}	Vecteur champ électrique
ϵ_0	Permittivité électrique du vide
ϵ_{av}	Erreur angulaire moyenne
$\epsilon_{av,x}$	Erreur angulaire sur θ_x
$\epsilon_{av,y}$	Erreur angulaire sur θ_y
ϵ_{mc}	Coût angulaire moyenne
ϵ_{ms}	Erreur quadratique moyenne
ϵ_r	Permittivité relative du matériau réflecteur
$\bar{\epsilon}_x$	Erreur absolue d'estimation de $MLPx$ faite sur un intervalle de 1° , intervenant dans le tracé des courbes d'erreur
f	Fréquence du signal
f_w	Fonction de pondération utilisée pour définir ϵ_{mc}
G	Gain en amplitude des antennes
Γ_v	Poids affecté aux DOA^r pour l'entraînement bisignal
$(\Gamma_v)_{k_1,k_2}$	Matrice de coefficients Γ_v pour l'entraînement bisignal mobile
h	Hauteur du sol par rapport au sol
I	Point d'incidence
J	Fonction coût
J_{sin}	Fonction coût sinusoïdale
λ	Longueur d'onde
n_{ix}	Bruit complexe additif à x_{ix}
N_e	Nombre d'époques d'entraînement
N_e^*	Nombre optimal d'époques d'entraînement
N_{ent}	Nombre de points d'entraînement
N_n	Nombre de neurones dans la couche cachée

N_n^*	Nombre optimal de neurones dans la couche cachée
N_{pre}	Nombre d'échantillons utilisés pour le pré-traitement
N_{post}	Nombre d'échantillons utilisés pour le post-traitement
N_{val}	Nombre de points de validation
P_m	Puissance moyenne du signal
\hat{p}	Vecteur unitaire de polarisation
ϕ	Angle d'azimut des coordonnées sphériques du signal direct
ϕ_r	Angle d'azimut des coordonnées sphériques de la réflexion
ψ	Angle d'incidence
ψ'	Angle d'élévation terrestre ($= 90^\circ - \psi$)
ψ_b	Angle de Brewster
R	Récepteur
R'	Point image de R par rapport au sol
S_p	Seuil de puissance de détection
S_p^t	Seuil de puissance de détection pour la phase d'entraînement
S_p^v	Seuil de puissance de détection pour la phase de validation
σ	Conductivité électrique du matériau réflecteur
σ^2	Variance de l'amplitude du bruit additif complexe
$\sigma_{av,y}$	Ecart-type moyen sur les estimations de θ_y
T_e	Taux d'erreur de classification
θ	Angle d'élévation des coordonnées sphériques du signal direct
θ_r	Angle d'élévation des coordonnées sphériques de la réflexion
θ_x	Angle entre la DOA du signal direct et l'axe X
$\theta_{x,r}$	Angle entre la DOA de la réflexion et l'axe X
$\bar{\theta}_x$	Valeur moyenne de θ_x obtenue par post-traitement
$\tilde{\theta}_x$	Estimation de θ_x par MLP_x

θ_y	Angle entre la DOA du signal direct et l'axe Y
$\theta_{y,r}$	Angle entre la DOA de la réflexion et l'axe Y
U_{ix}	Signaux théoriques désirés en sortie du combineur
$U_{x,i}$	Signaux complexes reçus par l'antenne i de l'axe X
\hat{u}_h	Vecteur unitaire de polarisation horizontale
\hat{u}_r	Vecteur unitaire de propagation de l'onde
\hat{u}_v	Vecteur unitaire de polarisation verticale
x_{ix}	Amplitude des signaux en sortie du combineur d'axe X
$x_{ix,r}$	Amplitude des signaux perturbés par une réflexion en sortie du combineur d'axe X
X_{ix}	Signaux de puissance en sortie du combineur d'axe X
$X_{ix,n}$	Signaux de puissance normalisés en sortie du combineur d'axe X
X_{ix}^b	Signaux de puissance bruités en sortie du combineur d'axe X
$X_{ix,n}^b$	Signaux de puissance bruités normalisés en sortie du combineur d'axe X
$X_{ix,r}^b$	Signaux de puissance bruités perturbés par une réflexion en sortie du combineur d'axe X
$X_{ix,r,n}^b$	Signaux de puissance bruités normalisés perturbés par une réflexion en sortie du combineur d'axe X
$\overline{X_{ix,n}^b}$	Valeur moyenne obtenue par pré-traitement sur les signaux de puissance bruités normalisés en sortie du combineur d'axe X
X_{max}	Signal d'intensité maximale reçu par le combineur sur l'ensemble du jeu de points
$\vec{X}_{x,n}$	Vecteur d'entrée de $MLPx$
\hat{x}	Vecteur unitaire de l'axe X
\hat{y}	Vecteur unitaire de l'axe Y

\hat{z} Vecteur unitaire de l'axe Z

SIGLES

<i>D</i>	Entraînement “sans réflexion”
<i>DR</i>	Entraînement “avec réflexion mono-configuration”
<i>DRM</i>	Entraînement “avec réflexion multi-configuration”
<i>DOA</i>	Direction Of Arrival
<i>EM</i>	Expectation Maximisation
<i>ESPRIT</i>	Estimation of Signal Parameters via Rotational Invariance Techniques
<i>HFSS</i>	High-Frequency Structure Simulator
<i>MLP</i>	Multi Layer Perceptron
<i>MLPd</i>	MLP de détection
<i>MLPx</i>	MLP d'axe X estimant θ_x
<i>MLPy</i>	MLP d'axe Y estimant θ_y
<i>MUSIC</i>	MUltiple SIgnal Classification
<i>p/r</i>	Par rapport
<i>RBF</i>	Radial Basis Function
<i>RNA</i>	Réseau de Neurones Artificiels
<i>RSB</i>	Rapport Signal à Bruit
<i>RF</i>	RadioFréquence
<i>SS</i>	Entraînement “bi-signal classique”
<i>SSD</i>	Entraînement “bi-signal décalé”
<i>SVM</i>	Support Vector Machine

LISTE DES TABLEAUX

TAB. 2.1	<i>Comparaison des résultats effectués sur des signaux bruités ($RSB = 20$ dB) avec et sans pré-traitement</i>	54
TAB. 3.1	<i>Comparaison des résultats effectués sur des signaux bruités perturbés par une réflexion, avec et sans pré-traitement . . .</i>	73
TAB. 3.2	<i>Résultats effectués sur des signaux bruités perturbés par une réflexion en utilisant une architecture à un seul MLP</i>	75
TAB. 3.3	<i>Résultats effectués sur des signaux bruités perturbés par une réflexion en utilisant des antennes à polarisation horizontale ou verticale, deux RNAs MLPx et MLPy</i>	78
TAB. 3.4	<i>Résultats effectués sur des signaux bruités perturbés par une réflexion en utilisant un réseau à deux paires d'antennes, deux RNAs MLPx et MLPy</i>	80
TAB. 3.5	<i>Comparaison de performances des deux MLPds</i>	88
TAB. 3.6	<i>Performances de MLPd optimal à 40 neurones entraîné pendant 45 époques</i>	89
TAB. 4.1	<i>Performances de l'entraînement sans réflexion testé sur MLPx et MLPy avec 30 neurones (N), entraînés pendant 30 époques (E)</i>	94
TAB. 4.2	<i>Performances de l'entraînement D testé sur MLPx et MLPy avec 40 et 50 neurones respectivement, entraînés pendant 30 époques, avec $\epsilon_r = 3.2$ et $\epsilon_r = 15$</i>	97
TAB. 4.3	<i>Performances de l'entraînement DR testé sur MLPx et MLPy avec 30 neurones, entraînés pendant 30 époques</i>	101
TAB. 4.4	<i>Performances de l'entraînement DR testé sur MLPx et MLPy avec 30 neurones, entraînés pendant 30 époques (d différente)</i>	101

TAB. 4.5	<i>Performances de l'entraînement DR testé sur MLP_x et MLP_y avec 30 neurones, entraînés pendant 30 époques (ϵ_r, différente)</i>	102
TAB. 4.6	<i>Valeurs numériques de Ω et d</i>	104
TAB. 4.7	<i>Performances de l'entraînement DRM testé sur (MLP_x, MLP_y) = (30N, 30E, 30N, 30E) et (MLP_x, MLP_y) = (40N, 30E, 50N, 30E)</i>	106
TAB. 4.8	<i>Performances de l'entraînement SS avec architecture optimisée pour $d = 10\text{ m}$ et $d = 100\text{ m}$</i>	112
TAB. 4.9	<i>Performances de l'entraînement SSD avec architecture optimisée pour $\epsilon_r = 3.2$ et $\epsilon_r = 15$</i>	116
TAB. 5.1	<i>Performances de l'entraînement D testé avec différents plans réflecteurs</i>	139
TAB. 5.2	<i>Performances de l'entraînement DRM testé avec différents plans réflecteurs et comparé avec l'entraînement D</i>	141
TAB. 5.3	<i>Performances de l'entraînement SSD testé avec différents plans réflecteurs et comparé avec l'entraînement D</i>	144
TAB. 5.4	<i>Performances des entraînements D et SSD, testés avec un plan réflecteur en asphalte (mesures en azimuth et élévation)</i>	149

CHAPITRE 1

INTRODUCTION

1.1 Problématique générale

Dans de nombreuses applications, la connaissance de la direction d'arrivée d'un faisceau sur un système est requise (problème de DOA : "direction of arrival" en anglais). Ce faisceau pourra contenir ou non de l'information utile :

- en télécommunications, le faisceau est un signal transportant des données et une bonne réception est nécessaire ; connaître la DOA du faisceau pourra alors servir à optimiser la qualité de réception.
- en poursuite automobile, l'onde émise représente l'information de position du véhicule de tête à suivre ; déterminer sa DOA au niveau du véhicule de queue est la première étape, et la plus ardue, d'une telle opération.

Dans les deux cas, la configuration spatiale du couple émetteur-récepteur est fonction du temps et un compromis entre précision et temps de calcul de l'estimation est à prendre en compte. Il faut garder à l'esprit qu'un système relativement peu précis mais rapide pourra résorber son handicap par son débit d'estimations. Ce sera d'autant plus avantageux que la DOA varie rapidement au cours du temps.

La première partie d'un tel système est constituée d'un réseau d'antennes qui capte l'onde incidente émise. C'est après, dans la partie en aval, que divergent les solutions, au niveau du traitement des signaux de sortie des antennes. Plusieurs méthodes existent et ont chacune leurs avantages et inconvénients.

- La plus répandue repose sur l'algorithme MUSIC (pour "Multiple Signal Classification"). Elle consiste à former la matrice d'autocorrélation spatiale basée sur les vecteurs de sortie du réseau d'antennes et à déterminer ses valeurs et vecteurs propres (El Zooghby, 1997), (Schmidt, 1986). C'est une méthode précise, même en présence de bruit, mais lourde en temps de calcul et inefficace lorsqu'il y a lieu des interférences avec une onde cohérente au signal incident, comme dans le cas d'une réflexion parasite. Un autre algorithme, dit ESPRIT (pour "Estimation of Signal Parameters via Rotational Invariance Techniques"), peut aussi être utilisé (Roy, 1989).
- L'algorithme EM (pour "expectation-maximization") utilise lui l'estimation du maximum de vraisemblance de la DOA du faisceau. Cette méthode itérative est laborieuse mais peut être efficace dans le cas d'interférences cohérentes (Feder, 1988).
- La séparation de sources est une autre technique utilisée. Elle est basée sur les différences de forme des signaux émis par les sources. Cette séparation devient difficile par contre lorsque les sources ne sont pas indépendantes, ce qui est le cas lorsqu'une réflexion perturbe le signal principal (Haykin, 1999).
- Une nouvelle voie utilise des réseaux de neurones artificiels (RNAs). Ce sont souvent des RBFs (pour "Radial Basis Function") dans la littérature (El Zooghby, 1997), (El Zooghby, 2000), (Lo, 1994, 1), (Lo, 1994, 2), (Wong, 1993). Ils exploitent l'information contenue dans la matrice d'autocorrélation spatiale comme MUSIC. Les RNAs réalisent une régression de la fonction liant l'espace des composantes de cette matrice à celui des DOAs. Ils sont rapides, idéaux pour des applications temps réel mais donnent des résultats moins précis en général que MUSIC. Par contre, avec un entraînement adéquat, il est possible de donner une estimation fiable même en présence d'une réflexion parasite.

Ce projet emploie cette technique. Il poursuit le travail déjà accompli par Michael Coudyser et Nelson Fonseca : à eux deux, ils ont mis en œuvre un système de DOA robuste au bruit. Les signaux traités par les RNAs sont combinés entre eux de manière à fournir une information non ambiguë. Ce sont des signaux de puissance, réels, exploitables par le réseau de neurones. Cette architecture est différente de la méthodologie classique basée sur la matrice d'autocorrélation spatiale. Elle repose sur une considération de moindre coût. C'est la ligne directrice de notre étude d'ailleurs : réaliser un système à faible coût (combinateur de puissance, nombre réduit d'éléments RF et d'antennes, RNAs, etc.) mais néanmoins performant.

De plus, des MLPs avaient été utilisés par Michael Coudyser. Leur entraînement aisé par rétropropagation d'erreur et leur souplesse d'emploi sous Matlab[®] en font encore les meilleurs candidats par rapport aux RBFs pour ce projet. Même si d'autres types de RNAs moins répandus sont le sujet de nombreux travaux et auraient donc pu être utilisés, on a préféré choisir parmi les approches les plus connues, i.e. entre les MLPs et les RBFs.

1.2 Revue de littérature

Il est important de comparer notre système aux autres études qui ont été développées, reposant sur des techniques différentes. On peut classer les différents papiers en deux familles : celles faisant l'estimation de la DOA en présence de signaux cohérents d'une part, et ceux la calculant sans interférences d'autre part.

Voyons ce dernier groupe pour commencer :

- Dans (Charpentier, 1999), un réseau de trois antennes disposées en triangle permet d'estimer l'azimut de la DOA sur une plage de 360° , en utilisant un MLP. Mais la précision d'un tel système est limitée à $\pm 18^\circ$ du fait des classificateurs

utilisés. Ce système est robuste au bruit et à une interférence non cohérente avec le signal principal. Surtout, des validations expérimentales corroborent les simulations.

- Dans (Lo, 1994, 1), un réseau de neurones RBF est utilisé avec un réseau uniforme de huit éléments pour estimer la DOA. Les simulations sont menées sous du bruit de RSB variable (de 0 à 40 dB), en présence d'un signal interférent, non corrélé et de même puissance que le signal principal. La résolution angulaire est de 2° . Aucune mesures expérimentales n'ont été faites.
- Des RBFs sont également utilisés dans (El Zooghby, 2000). Différents réseaux d'antennes, de 4 à 12 éléments, ont été testés, sous un bruit de RSB variant entre 5 et 10 dB typiquement. Un tel système permet de retrouver les DOAs de plusieurs sources non corrélées, même si le nombre de sources est supérieur au nombre d'antennes. La résolution angulaire est ici de 2° . Mais l'étude n'a pas dépassé le stade de simulations.
- Dans (Shieh, 2000), un réseau de 4 antennes est couplé à un réseau de neurones à logique floue. Différentes simulations ont été effectuées, sous un RSB variant de -5 à 25 dB. L'erreur quadratique moyenne obtenue est de 0.15° avec ce type de réseau sous un bruit de RSB = 20 dB, sans interférences d'aucune sorte. Ici encore, aucune mesures expérimentales n'ont été faites.
- La méthode présentée dans (Chen, 1994) n'utilise pas de RNAs mais des estimateurs de Kalman couplés à MUSIC pour trouver la DOA. La taille du réseau est assez importante (huit éléments) mais l'estimation faite en présence de bruit, même fort, est bonne : 0.1° d'erreur quadratique moyenne sous un RSB de 5 dB. Par contre, cette estimation est réalisée sans sources corrélées.

Notre étude porte sur le problème des interférences causées par une source cohérente avec le signal principal dont on veut estimer la DOA. Il est donc plus intéressant de regarder les travaux qui ont été réalisés dans cette direction :

- Dans (El Zooghby, 1997), l'estimation de la DOA se fait grâce à un réseau uniforme de 6 antennes complété par des RBFs. Le système est capable de retrouver les DOAs de deux signaux cohérents de même puissance. L'erreur angulaire est de l'ordre de plusieurs degrés, mais aucune précision sur le bruit n'est spécifiée. De plus, les résultats s'arrêtent aux simulations.
- Dans (Lo, 1994, 2), des RBFs sont également utilisés pour retrouver les DOAs de deux signaux cohérents, un signal direct et une réflexion générée par un lac. Le RSB est situé aux alentours de 20 dB. Les applications d'un tel système sont différentes des nôtres puisque leur cône d'intérêt est très resserré : le système estime les DOAs sous des incidences rasantes. Une précision inférieure à 0.1° est donc nécessaire et même obtenue expérimentalement. Mais pas moins de 32 antennes sont utilisées pour arriver à ce résultat.
- Dans (Ma, 1999), une estimation de la DOA dans une portion d'espace est cette fois réalisée. Un maillage carré de 4×4 antennes est utilisé. Différents niveaux de bruit sont simulés, de RSB variant de 4 à 20 dB. Un RNA à logique floue est employé pour retrouver la DOA du signal principal quand deux plans réflecteurs orthogonaux génèrent chacun une réflexion parasite (sol et mur par exemple). Plutôt que de perturber la mesure, les réflexions sont utilisées pour localiser la source. Les simulations sont concluantes, les erreurs très faibles mais ce système est opérationnel pour cette disposition particulière des plans réflecteurs. Cette méthode ne peut être utilisée dans un environnement qui n'est pas connu à l'avance. Son intérêt est donc limité.

1.3 Le problème des réflexions parasites

L'apport majeur de ce projet concerne sa robustesse à une réflexion parasite (qui est cohérente du signal principal). En effet, quelle que soit la future application, même

si un seul signal est utilisé, des réflexions de différentes intensités vont se produire dépendamment de l'environnement dans lequel évolue le porteur du système.

Comme point de départ, on a émis l'hypothèse que le réseau était monté sur une voiture, dans un cadre de poursuite automobile (voir Figure 1.1). Ici, la route est omniprésente et crée continuellement une réflexion du signal incident qui sera également captée par le système de DOA. La difficulté du problème est accrue par le fait que la forme (bosses, nids de poule, inclinaison, etc.) et les matériaux constitutifs (ciment, asphalte, graviers, terre, etc. recouverts d'eau ou non) de la route sont inconnus. De ce fait, la DOA ainsi que l'amplitude et la phase relatives du signal réfléchi sont toujours inconnues.

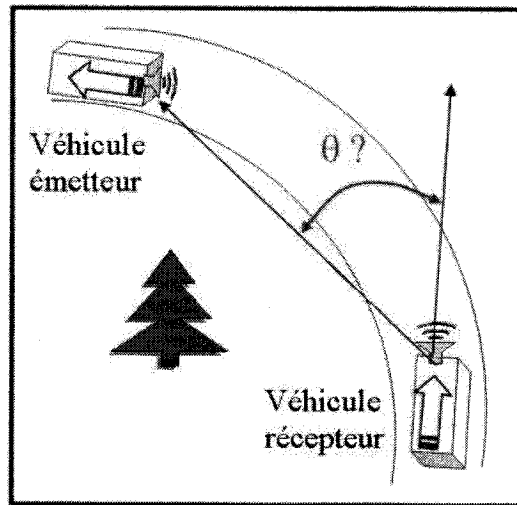


FIG. 1.1 *Problématique de la poursuite automobile*

La clef de voûte du projet réside dans un choix réfléchi des points d'entraînement de nos MLPs. En effet, en entraînant spécifiquement les RNAs, i.e. avec un signal principal et un signal réfléchi, il est possible de réduire significativement l'erreur angulaire de l'estimation faite par notre système en présence d'interférences et de

bruit, par rapport à un entraînement traditionnel basé sur un seul signal comme c'était le cas dans le mémoire de Michael Coudyser.

Pour finalement valider la structure neuronale, un modèle de réflexion sur route a été développé en simulations. Il nous a permis de choisir la configuration neuronale la plus adaptée (du point de vue nombre de neurones sur la couche cachée et nombre d'époques d'entraînement) à ce problème.

Enfin, des expériences ont été menées en chambre anéchoïde sur un vrai réseau d'antennes, fonctionnant à 77 GHz pour une question de compatibilité avec les systèmes anti-collisions de voitures déjà existant. Un morceau d'asphalte y était placé afin de simuler la réflexion parasite. Les mesures de sorties des antennes, en amplitude et phase, nous ont permis de générer des points de test plus proches de la réalité physique et de confirmer le travail accompli en simulations.

A la fin du travail, on souhaite que le système soit capable d'estimer la DOA d'un faisceau contenu dans un cône d'ouverture 90° en présence d'une réflexion parasite de caractéristiques inconnues et de bruit gaussien dont le RSB est supérieur à 20 dB, ce qui correspond à un niveau de bruit couramment rencontré dans la littérature (Lo, 1994, 2), (Shieh, 2000), (Ma, 1999). On essaiera de trouver différentes solutions afin d'obtenir la précision la plus grande possible.

1.4 Organisation du mémoire

Le mémoire s'articule de la manière suivante :

- Le chapitre 2 récapitule les travaux précédents accomplis par Michael Coudyser et Nelson Fonseca et relate les résultats qu'ils ont obtenus. Quelques améliorations y ont été apportées.

- Le chapitre 3 modélise l’effet d’une réflexion parasite issue d’une route et explique les modifications de structure induites sur le système, tant au niveau des antennes que des RNAs. Il reprend également diverses tentatives d’amélioration du système plus ou moins fructueuses.
- Le chapitre 4 montre différentes méthodes de génération des points d’entraînement pour essayer de déterminer la structure neuronale optimale du point de vue de l’entraînement, pour le problème considéré. Les erreurs d’estimation sont en effet différentes selon que l’on entraîne ou non les MLPs avec un ou deux signaux.
- Le chapitre 5 expose les résultats expérimentaux : il décrit le prototype et le déroulement des expériences. Ces résultats nous ont permis de valider notre modèle de réflexion parasite et de choisir le type d’entraînement le plus approprié en regard de la réflectivité du matériau.

CHAPITRE 2

SYNTHÈSE DES TRAVAUX PRÉCÉDENTS ET APPORTS PERSONNELS

2.1 Description générale du système

Un des systèmes de DOA mis au point par Michael Coudyser et Nelson Fonseca est celui décrit dans la Figure 2.1 et récapitulé dans ce chapitre.

Dans tout ce qui suit, on considère que le système ne reçoit qu'un seul signal incident, la réflexion n'est pas encore envisagée. Tout d'abord, deux réseaux d'antennes orthogonaux sont utilisés orientés selon deux axes perpendiculaires, X et Y . On utilise deux axes différents pour être capable d'estimer une DOA dans l'espace en trois dimensions. En effet, un seul réseau permet de retrouver l'angle d'arrivée d'une onde incidente référencé par rapport à l'axe du réseau d'antennes, qui est l'axe d'alignement des éléments du réseau. Pour cela, on utilise les déphasages entre les antennes du réseau, qui varient avec la DOA. Un réseau nous permettant de retrouver un angle entre deux axes, on a une infinité de solutions formant un cône solution. En prenant un autre réseau d'axes non colinéaires, il est possible de déterminer sans ambiguïté la véritable DOA en prenant l'intersection de ces deux cônes solution, qui est une droite, soit la DOA recherchée. Comme notre système final doit répondre à une considération de moindre coût, on s'est fixé un nombre maximal d'éléments pour chaque réseau, égal à trois. Un des problèmes dans la suite sera donc de trouver le réseau nous donnant les meilleurs résultats en terme de précision angulaire, entre celui ayant deux et l'autre ayant trois antennes. Dans (Coudyser, 2003), la solution à deux paires de réseaux de deux antennes avait été

retenue. Il est possible que celle-ci ne soit plus la meilleure quand on considère en plus une réflexion parasite dans le problème.

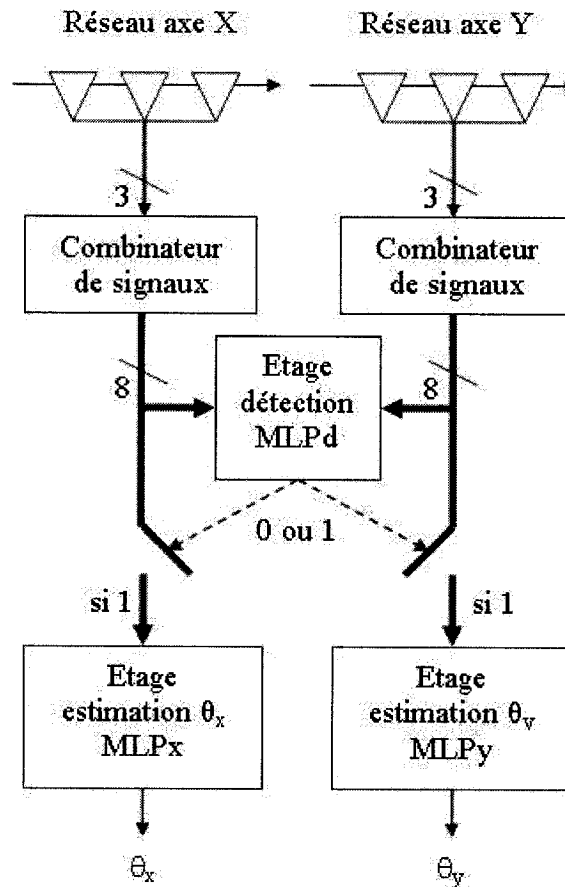


FIG. 2.1 Schéma du système global d'estimation de la DOA

On utilise ensuite un combineur de signaux par réseau d'antennes, soit deux en tout. Ils sont en fait constitués de lignes de transmission RF et de coupleurs hybrides. Ils prennent les signaux de sortie complexes de chacune des antennes, les mélangent entre elles afin d'obtenir en sortie des signaux de puissance de forme vou-

lue et optimisée pour faciliter la tâche aux RNAs. Chacun des combineurs génère ainsi 4 ou 8 signaux de puissance quand les réseaux sont constitués respectivement de 2 ou 3 antennes.

Ces signaux de puissance sont ensuite normalisés par rapport au plus fort de façon à rendre la réponse du système indépendante de l'atténuation du signal. Bien sûr, on se fixera un seuil de perception, à partir duquel on fera le travail d'estimation. Si les signaux sont trop faibles et inférieurs à ce seuil, "noyés" dans le bruit en quelque sorte, on décidera de ne pas chercher la DOA du signal, puisque celle-ci risque d'être très éloignée de la réalité. Avant d'être utilisés tels quels par les RNAs, les signaux de sortie des combineurs subissent aussi une conversion analogique-numérique.

La partie "intelligence artificielle" rentre en action à partir de ce moment par l'entremise des MLPs. Un premier étage est constitué d'un réseau de neurones à une couche cachée, dont les fonctions de transfert sont des sigmoïdes. La couche de sortie est formée par un neurone, dont la fonction de transfert est un échelon. Ce classificateur est appelé MLPd : à partir d'un vecteur de 8 ou 16 composantes pour des réseaux de 2 ou 3 antennes respectivement (vecteur qui est en fait composé par la donnée des signaux de puissance normalisés et numérisés des deux combineurs), ce MLP, une fois correctement entraîné, sera capable de nous dire si oui ou non le signal incident est compris dans le cône d'intérêt de demi-angle au sommet 45° . Si oui, sa sortie vaudra 1, sinon, elle sera à 0.

Si MLPd rend 1, les étages d'estimation MLPx et MLPy sont activés : chacun des deux MLPs détermine l'angle du signal incident par rapport à l'axe du réseau à partir de la donnée des signaux du combineur correspondant. Ces angles sont notés θ_x et θ_y et nous donnent sans ambiguïté la DOA du signal incident. Ils sont représentés dans la Figure 2.2. Si MLPd rend 0, aucun traitement spécifique n'est effectué : on sait déjà que le signal n'appartient pas au cône d'étude.

2.2 Structure du réseau d'antennes

2.2.1 Deux réseaux de trois antennes orthogonaux

Le système de réception est constitué d'une paire de réseaux de chacun trois antennes orientés selon deux axes orthogonaux, X et Y (voir Figure 2.2).

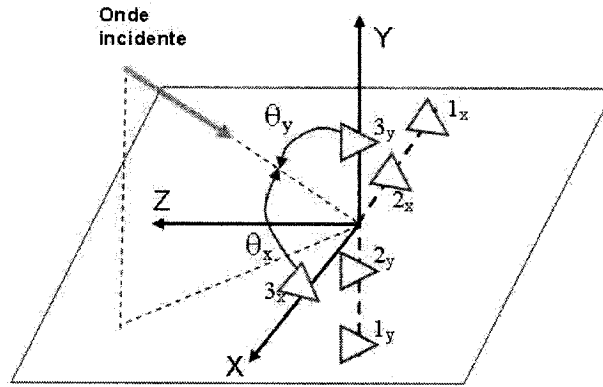


FIG. 2.2 Paire de réseaux d'antennes orthogonaux et définition des angles θ_x et θ_y

L'axe Z a été rajouté de telle façon à ce que le référentiel (X, Y, Z) forme un trièdre direct. L'axe Z est aussi l'axe principal d'observation, soit l'axe de révolution du cône d'étude. Si ce réseau était accroché au pare-chocs d'une voiture, l'axe Y serait vertical, X horizontal et Z pointerait vers l'avant de la voiture. Ce sont deux réseaux linéaires, i.e. les éléments sont alignés, mais non uniformes : l'espacement entre ces derniers n'est pas constant. Ces espacements ont été optimisés par Michael Coudyser de façon à générer des signaux de puissance de forme voulue en fonction de la DOA, en sortie des combinateurs de signaux. On a représenté ces distances inter-éléments dans la Figure 2.3 pour le cas à trois antennes par réseau toujours (pour le cas à deux antennes, voir (Coudyser, 2003)).

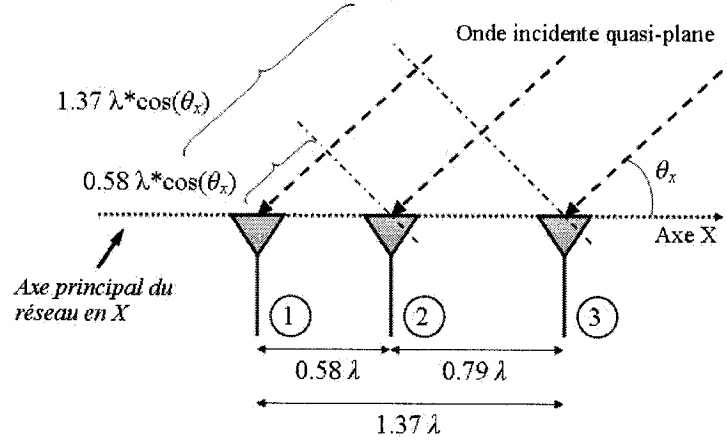


FIG. 2.3 Distance séparant les 3 antennes et déphasages associés

Les éléments sont identiques. Dans (Fonseca, 2003)), deux types différents d'antennes avaient été testés : des antennes spirale d'Archimède ou des antennes à patch carré alimentées par une fente en croix. Dans les deux cas, ce sont des antennes à polarisation circulaire. On verra dans la suite pourquoi dans ce mémoire, on a pu se contenter d'antennes à polarisation verticale, beaucoup plus simples à mettre en œuvre. Ces antennes seront aussi très peu directives, avec un gain typique en $\cos \theta$ pour les simulations avec θ l'angle formé entre l'axe Z et celui de la DOA.

2.2.2 Modélisation du faisceau d'arrivée et équations de sortie des antennes

On utilise ensuite la sortie complexe de chaque antenne pour alimenter le combineur de signaux. Les modèles adoptés dans les simulations pour générer les signaux de sorties sont présentés dans cette section. On considère que l'onde reçue par le système est de nature sphérique. On ne se préoccupe pas de sa polarisation car elle n'influence pas les résultats. Ce ne sera pas le cas par contre quand on considèrera

une réflexion parasite, dans le prochain chapitre. Le signal est à bande étroite, de fréquence principale fixée. Pour un point d'observation situé à une distance d de l'émetteur, l'onde incidente est de la forme :

$$A(d) = \alpha \frac{e^{-2\pi j d / \lambda}}{d} \quad (2.1)$$

où α est un nombre complexe constant et λ la longueur d'onde. λ est liée à la fréquence f du signal par la relation :

$$\lambda = \frac{c}{f} \quad (2.2)$$

On peut donc connaître quelle est la forme de l'onde au niveau des trois antennes (voir Figure 2.3). On suppose que localement, l'onde sphérique peut être considérée comme plane (approximation de l'onde quasi-plane, pour des sources lointaines, ce qui est le cas ici) au niveau des trois antennes. Le facteur de phase entre les trois antennes est différent et il est donné par la différence de chemin optique parcouru par l'onde le long du rayon lumineux (approximation de l'optique géométrique). On suppose que l'antenne 1 est située à une distance d_1 de la source. Comme les équations sont symétriques pour les réseaux d'axe X et Y , seules les équations en X sont exposées :

$$U_{x1} = \alpha \frac{e^{-2\pi j d_1 / \lambda}}{d_1} \quad (2.3a)$$

$$U_{x2} = \alpha \frac{e^{-2\pi j d_1 / \lambda}}{d_1} \exp(2\pi j \cdot 0.58 \cdot \cos \theta_x) \quad (2.3b)$$

$$U_{x3} = \alpha \frac{e^{-2\pi j d_1 / \lambda}}{d_1} \exp(2\pi j \cdot 1.37 \cdot \cos \theta_x) \quad (2.3c)$$

U_{x1} , U_{x2} et U_{x3} sont respectivement les signaux reçus par les antennes 1, 2 et 3. Ces antennes ont théoriquement le même facteur d'élément, i.e. le même gain, noté G et fonction de θ (ou de θ_x et θ_y). On a aussi facilement les signaux de sortie de

chacune des antennes :

$$A_{xi} = G(\theta) U_{xi}, i = 1, 2, 3 \quad (2.4)$$

Pour nos simulations, on utilise $G(\theta) = \cos \theta$. Les relations de passage entre les angles sphériques θ et ϕ d'une part, et les angles θ_x et θ_y d'autre part, sont donnés dans la suite :

$$\cos \theta_x = \sin \theta \cos \phi \quad (2.5a)$$

$$\cos \theta_y = \sin \theta \sin \phi \quad (2.5b)$$

$$\cos \theta = \sqrt{1 - (\cos^2 \theta_x + \cos^2 \theta_y)} \quad (2.5c)$$

L'équation 2.5c exprime le gain des antennes en fonction des angles θ_x et θ_y . Elle s'obtient directement à partir des deux équations 2.5a et 2.5b. $\theta \in [0^\circ, 90^\circ]$ et θ_x ou $\theta_y \in [0^\circ, 180^\circ]$. En effet, le demi-espace considéré dans notre étude est celui défini par $z > 0$. Ce dernier inclut bien évidemment le cône d'intérêt pour le calcul de la DOA, celui d'ouverture 90° autour de l'axe Z . La Figure 2.4 définit ces angles pour une DOA particulière.

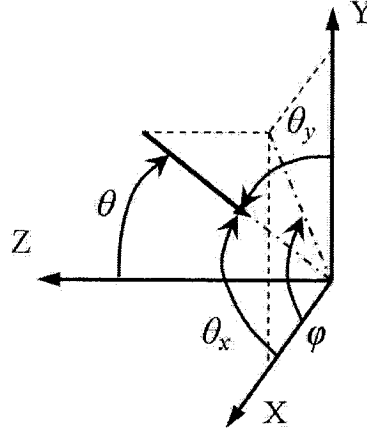


FIG. 2.4 Définition des angles du problème

2.3 Choix du combineur de signaux

2.3.1 Choix des signaux théoriques

Les trois signaux de sortie des antennes sont ensuite utilisés dans le combineur de signaux pour générer des signaux de forme voulue. Le but de ce processus est de réduire au maximum les ambiguïtés entre différentes DOAs dans le cône d'étude. Les angles θ_x et θ_y varient dans l'intervalle $[45^\circ, 135^\circ]$ pour le cône considéré.

On ne peut pas utiliser tels quels les six signaux de sortie des antennes, en extraire les puissances et entraîner un RNA avec ceux-ci parce que ces puissances sont les mêmes. Utiliser les signaux de sortie complexes des antennes, i.e. leur amplitude et leur phase, est beaucoup plus difficile à mettre en œuvre qu'un système basé sur les puissances. En effet, le déphasage entre les antennes permet de remonter à la valeur de la DOA mais cette méthode nécessite l'emploi de comparateurs de phase dont le paramétrage et la synchronisation sont très exigeants (Coudyser, 2003)).

L'idée consiste donc à utiliser d'autres signaux de puissance, de forme voulue, mais qui réduisent les ambiguïtés possibles entre des DOAs différentes. Les quatre signaux théoriques que l'on veut générer pour chacun des deux combineurs ont soit la forme de fonctions en créneaux, soit la forme de fonction en dents de scie, quand on les exprime en fonction de θ_x ou θ_y . Ces fonctions théoriques désirées sont représentées sur la Figure 2.5.

Les fonctions U_1 et U_2 sont des fonctions binaires (ne prennent que deux valeurs possibles) qui fragmentent le cône d'étude en 16 secteurs différents (quatre secteurs en θ_x et quatre en θ_y , d'ouverture 22.5°). On peut donc dans un premier temps établir dans quelle section du cône se situe la DOA en considérant ces quatre signaux de présence.

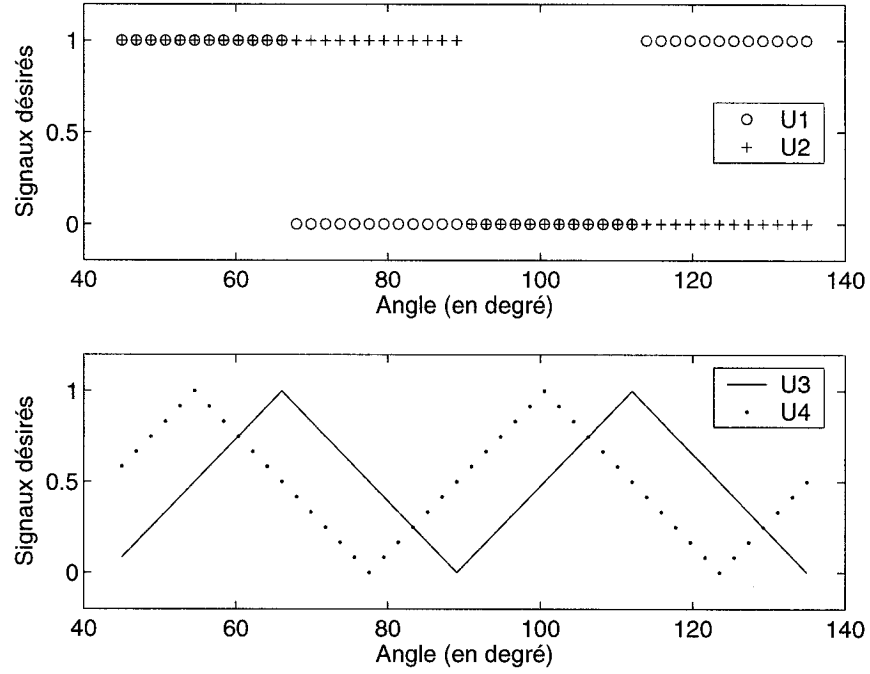


FIG. 2.5 *Signaux théoriques de présence et de précision en fonction de θ_x ou θ_y*

Les autres signaux sont en dents de scie et permettent d'affiner l'estimation : sachant dans quel fragment se situe la DOA, en consultant ces signaux, on peut estimer précisément l'angle d'arrivée du faisceau incident. Tout ce traitement sera effectué par les MLPs. La Figure 2.6 représente la sectorisation du cône d'étude. Le disque représenté est en fait une représentation en vue de dessus de la calotte sphérique représentant notre cône d'intérêt. Elle est définie de cette façon : on considère dans un premier temps une sphère unité dans notre système d'axe cartésien (X, Y, Z) . Le cône d'intérêt, d'axe principal Z et d'ouverture 90° , coupe cette sphère selon un cercle et définit deux zones : notre calotte sphérique, située dans le cône et le reste de la sphère situé à l'extérieur du cône. En représentant la calotte en vue de dessus (pour un observateur situé le long de l'axe Z), elle se réduit au disque de la Figure 2.6.

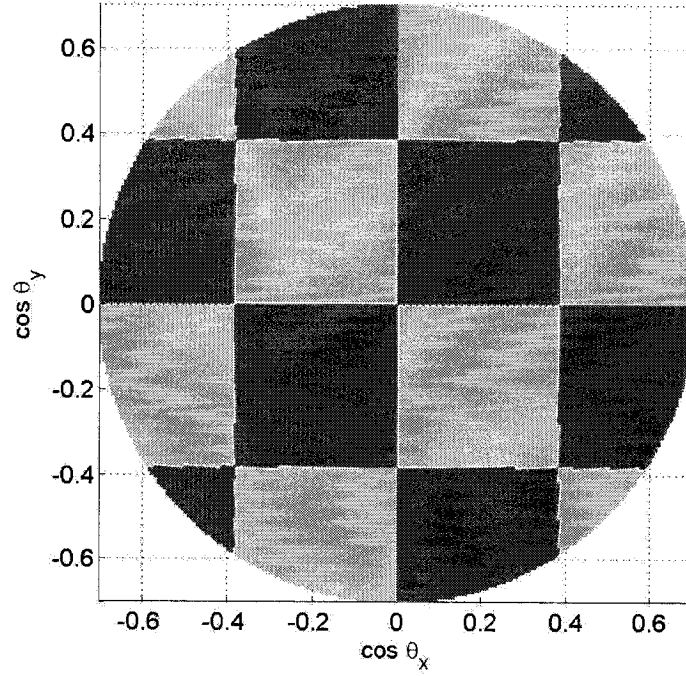


FIG. 2.6 *Sectorisation du cône d'étude en 16 parties, représentée à l'aide d'une portion de sphère unité en vue de dessus*

Selon la DOA, i.e. selon les angles θ_x et θ_y , les fonctions U_{1x} , U_{2x} , U_{1y} et U_{2y} sont soit positives, soit négatives, et permettent de savoir dans quelle case de la calotte se situe la DOA. Les quatre autres signaux, U_{3x} , U_{4x} , U_{3y} et U_{4y} , affinent et achèvent le travail d'estimation.

2.3.2 Optimisation des signaux réels

La tâche principale consiste à établir l'architecture du combinateur pour générer des signaux réels (à partir des signaux de sortie des antennes) les plus proches possibles des signaux théoriques précédents. On les appelle $X_i, i = 1..4$. On peut

montrer (voir (Coudyser, 2003)) que chaque signal X_i peut être exprimé en fonction de la DOA par une relation de la forme (en supposant des antennes identiques omnidirectionnelles, i.e. de gain unité pour toutes les directions de l'espace) :

$$X_i(\theta_{x,y}) = 1 + \cos\left(\alpha_i + \frac{2\pi\delta_i}{\lambda} \cos \theta_{x,y}\right) \quad (2.6)$$

$\theta_{x,y}$ est soit θ_x , soit θ_y , α_i est une phase additionnelle et δ_i est la distance entre une paire d'antennes. Il suffit ensuite de déterminer les α_i et les δ_i permettant aux fonctions X_i de se rapprocher le plus possible des fonctions U_{j_i} . Mathématiquement, il suffit de minimiser les quatre fonctions f_i (données dans l'équation 2.7) par rapport à α_i et δ_i .

$$f_i(\delta_i, \alpha_i) = \int_{\frac{\pi}{4}}^{\frac{3\pi}{4}} |X_i(\theta_{x,y}) - U_{j_i}(\theta_{x,y})| d\theta_{x,y} \quad (2.7)$$

Cette optimisation nous donne quatre valeurs pour α_i et quatre autres pour δ_i (pas de différences entre θ_x et θ_y vu la symétrie du problème). Ces résultats sont consignés dans (Coudyser, 2003) et aboutissent d'une part aux distances entre éléments du réseau données précédemment et d'autre part à l'élaboration du combineur de signaux final représenté dans la Figure 2.9.

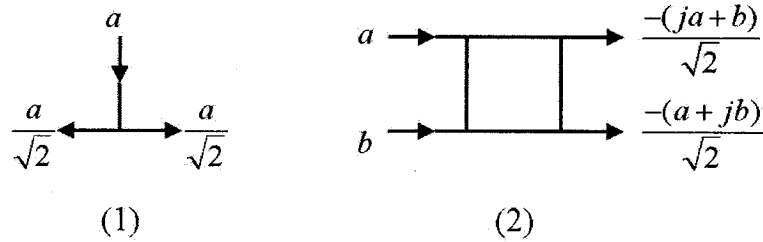


FIG. 2.7 Formulation théorique d'un diviseur de puissance (1) et d'un coupleur hybride 90° (2)

Le combineur est constitué de lignes de transmissions RF, de diviseurs de puissance et de coupleurs hybrides 90° . La Figure 2.7 redonne la formulation théorique d'un diviseur de puissance et d'un coupleur hybride 90° classique (Fonseca, 2003).

La définition du coupleur hybride 90° utilisée dans les blocs de la Figure 2.9 est légèrement différente de la définition de coupleur hybride donnée dans la Figure 2.7. Il suffit de rajouter un déphasage de 90° sur la première entrée d'un coupleur hybride 90° classique pour obtenir la formulation théorique des coupleurs hybrides formant le combineur de signaux. La Figure 2.8 en donne la formulation théorique.

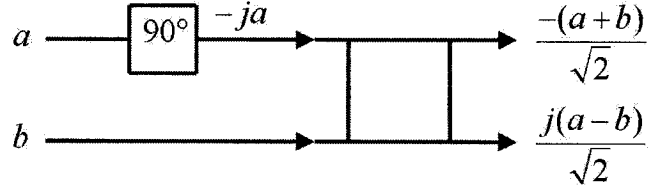


FIG. 2.8 Formulation théorique du coupleur hybride 90° réellement utilisé dans le combineur

Les signaux de puissance réels optimisés sont X_1, X_4, X_5, X_7 . L'opération consistant à prendre la puissance des signaux en sortie des coupleurs hybrides est implicite et n'est pas représentée sur la Figure 2.9. On remarque qu'avec les coupleurs hybrides, on génère deux fois plus de signaux que désiré. On a le choix de les utiliser ou non par la suite pour entraîner nos RNAs. Les équations 2.8 expriment les huit signaux X_{ix} en fonction des signaux de sortie des antennes du réseau d'axe X (les mêmes pour l'axe Y).

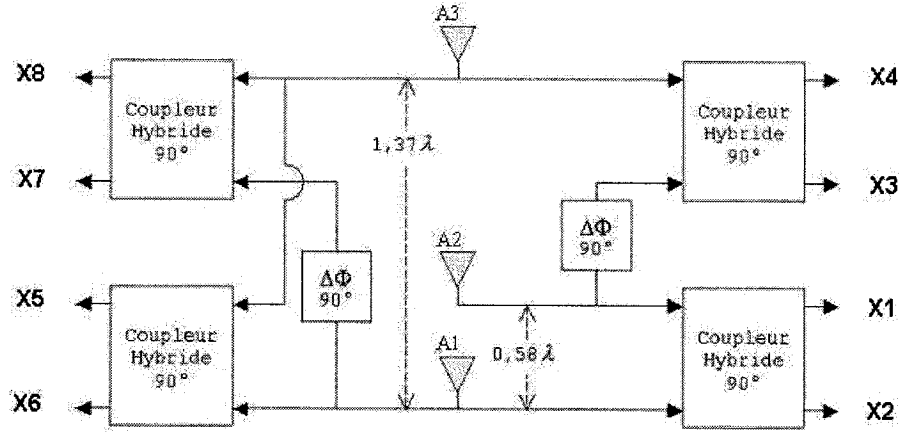


FIG. 2.9 Schéma du combineur de signaux de puissance optimal

$$X_{1x} = \left\| -j \frac{A_{x1} - A_{x2}}{2} \right\|^2 \quad (2.8a)$$

$$X_{2x} = \left\| -\frac{A_{x1} + A_{x2}}{2} \right\|^2 \quad (2.8b)$$

$$X_{3x} = \left\| -\frac{jA_{x2} + A_{x3}}{2} \right\|^2 \quad (2.8c)$$

$$X_{4x} = \left\| -\frac{A_{x2} + jA_{x3}}{2} \right\|^2 \quad (2.8d)$$

$$X_{5x} = \left\| -j \frac{A_{x1} - A_{x3}}{\sqrt{8}} \right\|^2 \quad (2.8e)$$

$$X_{6x} = \left\| -\frac{A_{x1} + A_{x3}}{\sqrt{8}} \right\|^2 \quad (2.8f)$$

$$X_{7x} = \left\| -\frac{jA_{x1} + A_{x3}}{\sqrt{8}} \right\|^2 \quad (2.8g)$$

$$X_{8x} = \left\| -\frac{A_{x1} + jA_{x3}}{\sqrt{8}} \right\|^2 \quad (2.8h)$$

On remarque que les signaux $X_i, i = 5..8$ sont de puissance deux fois plus petite que les signaux $X_i, i = 1..4$. Ceci s'explique par le tracé (nombre de divisions) des lignes de transmission. La Figure 2.10 représente les quatre signaux réels obtenus X_1, X_4, X_5, X_7 , comparés aux quatre signaux théoriques désirés.

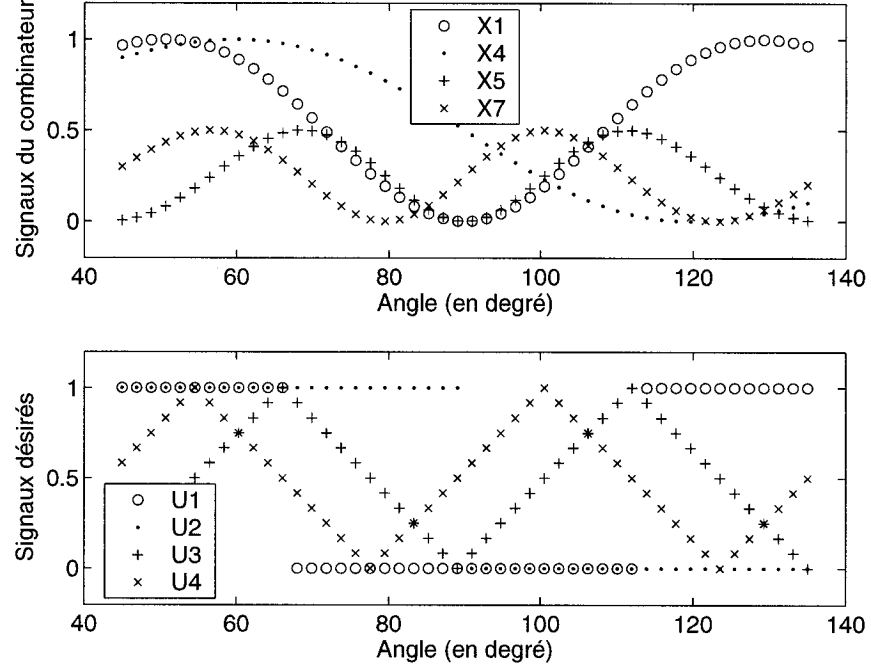


FIG. 2.10 *Signaux de sortie du combineur et signaux théoriques désirés en fonction de θ_x ou θ_y*

2.3.3 Utilisation des huit signaux de sortie du combineur

Il est également utile de représenter les quatre signaux restants pour savoir s'il vaut la peine qu'on les utilise dans la suite pour entraîner nos RNAs (voir Figure 2.11). On le fera s'ils complètent bien les signaux X_1, X_4, X_5, X_7 de manière à obtenir une redondance d'informations telle qu'elle augmente la robustesse de notre système de

détection (robustesse au bruit et à une réflexion parasite en l'occurrence).

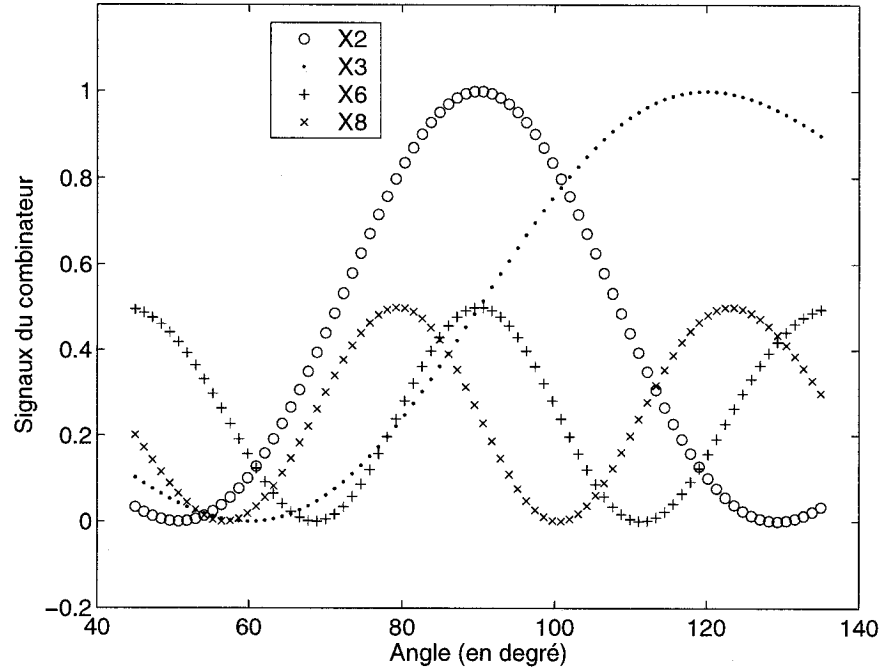


FIG. 2.11 *Signaux complémentaires du combineur en fonction de θ_x ou θ_y*

On remarque que les signaux X_2, X_3, X_6, X_8 sont en fait les signaux X_1, X_4, X_5, X_7 , mais décalés. Il est donc judicieux ici de les utiliser dans notre problème de DOA. On passe donc d'un vecteur de six signaux complexes regroupant les sorties des antennes des deux réseaux à un vecteur réel de 16 composantes, celles en sortie des deux combineurs, pour estimer deux angles, θ_x et θ_y . Cette hausse de dimensionnalité est souvent utilisée en classification : un problème complexe de classification de motifs projeté non linéairement dans un espace de grande dimension, a plus de chance d'être linéairement séparable que dans un espace de petite dimension. Ceci constitue le théorème de Cover (Haykin, 1999).

2.3.4 Définition d'un seuil de perception pour l'estimation de la DOA

Selon l'intensité des signaux en sortie du combineur, directement dépendante de celle du ou des faisceaux captés par le réseau d'antennes, on fera le travail d'estimation ou non. Il ne sert à rien d'estimer une DOA si l'on sait que cette mesure est noyée dans le bruit parce que le signal reçu est trop faible. Ce seuil est à définir en accord avec la qualité du matériel utilisé dans la fabrication future du système (qualité des antennes, des lignes de transmission RF, du combineur de signaux, etc.)

2.3.5 Normalisation et numérisation des vecteurs d'entrée des RNAs

Les signaux de sortie du combineur ne sont pas exactement ceux que l'on va fournir aux RNAs. En effet, ces signaux sont analogiques, alors que les RNAs acceptent généralement des signaux numériques. Ce sera en tout cas probablement le cas dans une future mise en œuvre. Une conversion analogique-numérique est donc nécessaire à ce stade, au cours de laquelle on normalise tous les signaux. En effet, pour que le système fonctionne dans des situations diverses, il faut le rendre indépendant de la puissance reçue par les antennes, qui est inconnue *a priori*. Une astuce simple mais néanmoins efficace consiste à diviser chacun des huit signaux de sortie de chaque combineur par le plus grand des huit, le signal de référence (ce dernier est différent pour chaque DOA). Les deux équations 2.9 résument le procédé, pour les signaux issus des deux combineurs, en X et en Y :

$$X_{ix,n}(\theta_x, \theta_y) = \frac{X_{ix}(\theta_x, \theta_y)}{\max_{i=1..8} X_{ix}(\theta_x, \theta_y)} \quad (2.9a)$$

$$X_{iy,n}(\theta_x, \theta_y) = \frac{X_{iy}(\theta_x, \theta_y)}{\max_{i=1..8} X_{iy}(\theta_x, \theta_y)} \quad (2.9b)$$

On appelle donc ici $X_{ix,n}$ et $X_{iy,n}$ les signaux normalisés et numérisés qui sont fournis aux différents RNAs : MLPd, MLPx et MLPy. Ils sont représentés sur la Figure 2.12.

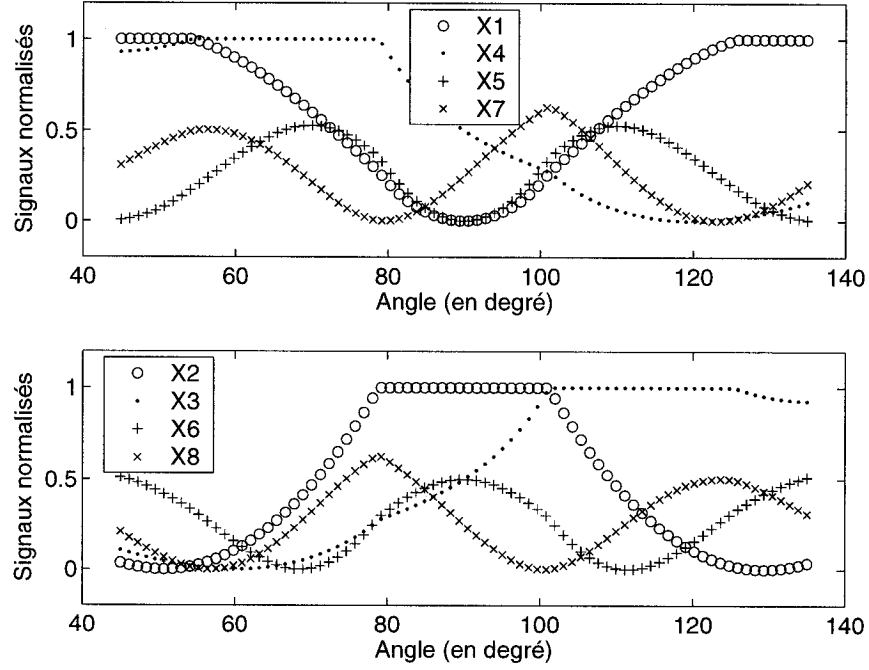


FIG. 2.12 Signaux normalisés en fonction de θ_x ou θ_y

Sont tracés plus hauts les huit signaux normalisés $X_{ix,n}(\theta_x)$ (ou $X_{iy,n}(\theta_y)$). En fait, ils sont fonctions à la fois de θ_x et θ_y du fait que le gain des antennes n'est pas isotrope. Il faut garder en tête que ce sont des signaux $X_{ix,n}(\theta_x, \theta_y)$ et $X_{iy,n}(\theta_x, \theta_y)$. On a plus exactement tracé $X_{ix,n}(\theta_x, \theta_y = 90^\circ)$ et $X_{iy,n}(\theta_x = 90^\circ, \theta_y)$. Mais la dépendance de $X_{ix,n}$ en θ_y (ou de $X_{iy,n}$ en θ_x) est faible grâce à la normalisation, si bien que l'estimation de θ_x faite par MLPx en lisant les huit signaux $X_{ix,n}$ (ou de θ_y faite par MLPy en lisant les huit signaux $X_{iy,n}$) reste bonne. On voit donc ici un des avantages d'utiliser des RNAs. Ils s'adaptent aux signaux qu'on leur fournit

même si les équations mathématiques ne sont pas respectées pour donner tout de même une bonne estimation de θ_x ou θ_y .

2.4 Description et fonctionnement des MLPs utilisés

Cette partie décrit le fonctionnement d'un neurone, puis d'un RNA et d'un MLP plus particulièrement (dans le cas général, puis dans le cadre de notre application), et explique leur utilité au sein du système de détection.

2.4.1 Qu'est-ce qu'un neurone artificiel ?

Les neurones artificiels sont directement inspirés de la biologie et de leurs homologues organiques. Ces derniers sont constitués d'un axone se terminant par une arborescence au niveau des synapses. Si l'ensemble des signaux synaptiques reçus par le neurone dépasse une certaine valeur, i.e. un seuil, alors le signal est transmis à la prochaine transition synaptique (en simplifiant). C'est ce principe qui est adopté et modélisé dans la Figure 2.13.

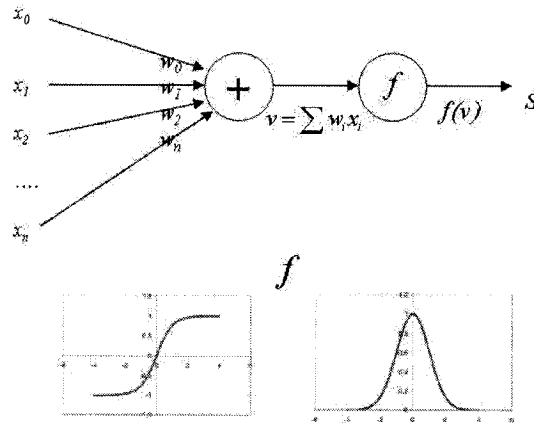


FIG. 2.13 *Modèle de neurone artificiel*

Les différentes dendrites sont représentées chacune par une valeur d'entrée x_i , réelle. L'ensemble de ces valeurs forme le vecteur d'entrée du neurone. Il est utile de rajouter un biais aux valeurs d'entrée. Celui-ci est en fait représenté par la composante x_0 fixée à 1. Le biais vaut alors w_0 . A la prochaine étape, chaque composante du vecteur d'entrée est multipliée par un poids synaptique w_i . Il permet de pondérer l'importance de chacune des entrées. Le neurone en tant que tel somme alors ces entrées pondérées entre elles. Ce résultat v est passé à travers une fonction d'activation f qui peut prendre des formes très diverses : une fonction linéaire, une fonction seuil, une sigmoïde, une gaussienne, etc. La sortie du neurone est alors $f(v)$. En se donnant une fonction d'activation, on peut modifier la relation entrée/sortie du neurone en jouant sur les poids synaptiques w_i . C'est l'optimisation de ces poids qui forme le travail de l'ingénieur en réseaux de neurones. La prochaine sous-partie montre comment, en interconnectant plusieurs neurones entre eux, il est possible de réaliser des tâches mathématiques complexes et variées via des RNAs.

2.4.2 Regroupement des neurones en réseaux

Un RNA est constitué de plusieurs neurones interconnectés entre eux. La façon dont ils sont reliés, les fonctions d'activations qui les composent, définissent différentes classes de réseaux. Celle qui nous intéresse ici, parce que c'est celle qui est utilisée dans notre système, est la famille des MLPs, pour des raisons pratiques : ce sont les réseaux de neurones les plus classiques et les mieux décrits sous Matlab[®] via la Neural Network Toolbox. Les neurones s'empilent par couches et les sorties des neurones de la couche $n - 1$ forment les vecteurs d'entrée des neurones de la couche n . Plus le nombre de couches du MLP est important, plus il sera capable de générer des fonctions sophistiquées. On se contentera ici d'un modèle de MLP à trois couches : celle d'entrée, celle de sortie et une couche cachée intercalée entre les deux ("cachée" parce qu'elle n'est pas visible de l'extérieur). La couche d'entrée est en

fait constituée par le vecteur d'entrée. Les neurones de la Figure 2.13 entrent dans la constitution de la couche cachée ainsi que de la couche de sortie. La Figure 2.14 représente un tel MLP.

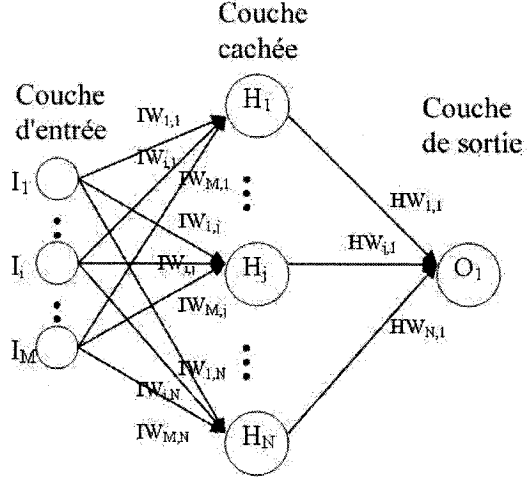


FIG. 2.14 Schéma d'un MLP classique à une couche cachée

Considérons la première couche : elle est constituée du vecteur d'entrée. Chaque cellule $I_i, i = 1, \dots, M$ est en fait une composante du vecteur d'entrée. Dans notre application, ce vecteur d'entrée est constitué des signaux de sortie numérisés des combinateurs. MLPx et MLPy prennent ainsi les huit signaux X_{ix} et X_{iy} respectivement, tandis que MLPd prend les 16 signaux au complet (X_{ix} et X_{iy}).

Chaque composante I_i est alors multipliée par un poids synaptique IW_{ij} au niveau du j^{eme} neurone H_j de la couche cachée. Les fonctions d'activation des neurones de la couche cachée sont des sigmoïdes en tangente hyperbolique dont la formule générale est donnée par l'équation 2.10. Elles sont dénommées “tansig” sous Matlab®.

$$tansig : x \rightarrow \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2.10)$$

Les sorties de chaque neurone H_j de la couche cachée sont à leur tour les entrées du neurone de sortie O , après avoir été pondérées par les coefficients HW_{j1} . La sortie sera l'angle θ_x (respectivement θ_y) pour MLPx (respectivement MLPy), et un nombre binaire (0 ou 1) pour MLPd. La fonction de transfert est linéaire pour MLPx et MLPy (fonction "purelin" de Matlab[®]). Pour MLPd, la fonction de transfert est une fonction seuil (fonction "hardlim" de Matlab[®]). La Figure 2.15 représente les trois fonctions de transfert apparaissant dans notre problème.

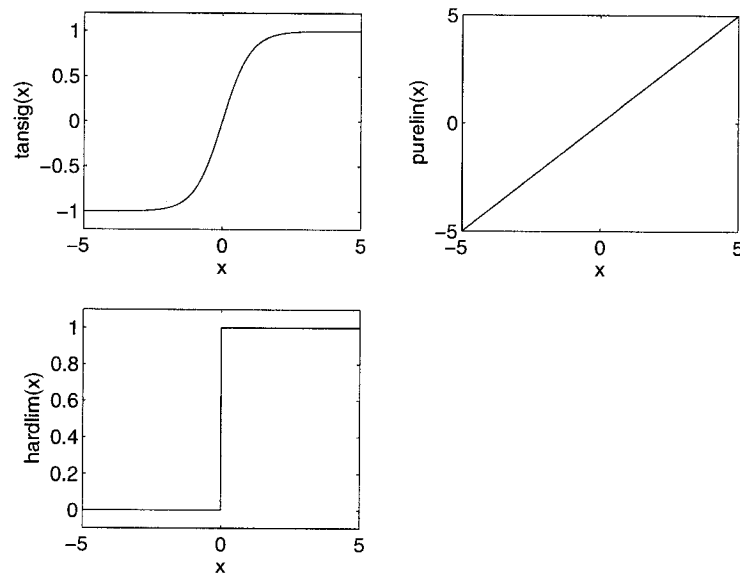


FIG. 2.15 Fonctions de transfert rencontrées : en sigmoïde, linéaire et en seuil

2.4.3 Principe de fonctionnement d'un MLP

On utilise des MLPs pour faire des régressions de fonctions. En effet, on cherche un moyen pour obtenir la relation existant entre les signaux de sortie des combineurs et la DOA. Celle-ci est loin d'être linéaire et il n'existe pas de forme analytique

pour exprimer la fonction $(X_{1x,n}, \dots, X_{8x,n}, X_{1y,n}, \dots, X_{8y,n}) \rightarrow (\theta_x, \theta_y)$. Utiliser des RNAs est un moyen très pratique d'obtenir une régression de cette fonction.

Pour que le RNA approxime la fonction désirée, on a besoin de connaître des couples (entrée, sortie) connus. Par un algorithme d'apprentissage décrit dans la suite, il est possible de modifier les poids synaptiques des neurones des différentes couches de façon à minimiser l'erreur entre les sorties théoriques, connues, et celles effectivement rendues par le RNA. Le critère à minimiser est souvent l'erreur quadratique moyenne par exemple. Une fois cette phase d'entraînement terminée, le réseau est prêt à être utilisé. Cette fois, en présentant un vecteur d'entrée inconnu, le RNA est capable de donner une bonne approximation de la sortie correspondante. Pour savoir si un RNA a été correctement entraîné, une méthode simple mais néanmoins efficace consiste à scinder le jeu de points connus en deux : un lot (le plus grand) qui servira à entraîner le réseau et un autre dit de validation (plus petit et différent du premier) dont le rôle est de comparer les sorties rendues par le RNA avec celles, connues, du lot de validation. Plus l'erreur est petite, meilleure est la qualité d'entraînement. Cette méthode s'appelle la validation croisée.

Le paragraphe suivant résume l'algorithme d'apprentissage d'un MLP (voir (Coudyser, 2003) pour plus de détails). Etant donné un jeu de points d'entraînements, on modifie les poids synaptiques (mis sous forme d'un vecteur $\vec{w} = (w_1, \dots, w_k, \dots, w_K)$ avec K le nombre total de poids synaptiques dans le MLP) à chaque itération de manière à réduire une fonction coût $J(\vec{w})$. Elle est définie en sommant les erreurs $L(\vec{w}, e_i)$ sur chaque échantillon $e_i, i = 1, \dots, N$ du jeu d'entraînement, en supposant que l'on a N échantillons :

$$J(\vec{w}) = \frac{1}{N} \sum_{i=1}^N L(\vec{w}, e_i) \quad (2.11)$$

Les poids synaptiques sont initialisés au début de l'algorithme et on se place à

l'étape n . Les poids valent alors $\vec{w}(n)$ et la fonction erreur $L(\vec{w}(n))$ (pour un échantillon donné). Si on appelle $\vec{y}(\vec{w}(n))$ la sortie réelle du MLP à l'étape n pour cet échantillon et \vec{t} la sortie théorique désirée (dans le cas général où la couche de sortie est constituée de plusieurs neurones, d'où les vecteurs) et si on prend l'erreur quadratique moyenne pour définir la fonction coût, on a :

$$L(\vec{w}(n)) = \frac{1}{2}(\vec{t} - \vec{y}(\vec{w}(n)))^2 \quad (2.12)$$

Pour minimiser $L(\vec{w})$, on modifie \vec{w} selon la méthode de descente du gradient :

$$\Delta w_i = -\eta \frac{\partial L}{\partial w_i} \quad (2.13)$$

Le paramètre η s'appelle le taux d'apprentissage. On obtient alors la mise à jour de \vec{w} pour l'étape $n + 1$:

$$\vec{w}(n + 1) = \vec{w}(n) + \Delta \vec{w}(n) \quad (2.14)$$

Pour améliorer la vitesse de convergence de l'algorithme, il est possible d'optimiser la valeur de $\eta(n)$ à chaque itération. En pratique, on utilise la fonction "trainlm" de Matlab[®] qui est basée sur l'optimisation de Levenberg-Marquart. Elle réside sur le calcul de l'inverse de la dérivée seconde de $L(\vec{w})$ qui est en fait la matrice Hessienne H . Soit η^* le paramètre d'apprentissage optimal :

$$\eta^* = H^{-1} \quad (2.15)$$

En pratique, l'inversion de H pose souvent des problèmes. L'optimisation de Levenberg-Marquart en calcule une estimation, H^{LM} :

$$H^{LM} = (H + \mu I)^{-1} \quad (2.16)$$

I est la matrice identité et μ est un coefficient à minimiser dépendamment des valeurs propres de H (Coudyser, 2003).

Les corrections à apporter à chaque poids synaptique ne sont pas calculées de la même façon selon son appartenance à telle ou telle couche du MLP. En fait, les modifications des poids de la couche j dépendent de données de la couche $j + 1$. Cet algorithme porte le nom de rétropropagation d'erreurs pour cette raison : on part de la sortie et on remonte vers l'entrée par couches successives. Cet algorithme est très bien détaillé dans (Haykin, 1999).

2.5 Améliorations proposées par rapport aux travaux de Coudyser

2.5.1 Choix des points d'entraînement

Le prochain problème qui s'offre à nous une fois adoptés l'architecture du combineur de signaux et le type de RNAs utilisés, est celui du choix des points d'entraînements de nos MLPs. Pour se faire, on a besoin de sélectionner des couples (entrée, sortie) : on veut générer des vecteurs de 16 composantes regroupant les signaux $X_{ix,n}$, $i = 1, \dots, 8$ et $X_{iy,n}$, $i = 1, \dots, 8$ pour un nombre important de DOAs.

Considérons dans un premier temps et à titre d'exemple l'entraînement de MLPx. Dans (Coudyser, 2003), les DOAs étaient choisies au hasard, et dans le cône d'intérêt d'ouverture 90° . Les paragraphes qui suivent décrivent le cheminement pour arriver à une méthode plus efficace.

2.5.1.1 Maillage arithmétique en θ et ϕ

La première idée qui nous vient à l'esprit consiste à considérer les angles sphériques θ et ϕ comme des suites arithmétiques, θ_n et ϕ_m , dont les caractéristiques sont données par les équations 2.17 et 2.18.

$$\Delta\theta = \frac{\pi}{4(N+1)} \quad (2.17a)$$

$$\theta_0 = \Delta\theta \quad (2.17b)$$

$$\theta_N = \frac{\pi}{4} \quad (2.17c)$$

$$\theta_n = \theta_{n-1} + \Delta\theta \quad (2.17d)$$

θ_n est une suite de $N+1$ éléments.

$$\Delta\phi = \frac{2\pi}{M+1} \quad (2.18a)$$

$$\phi_0 = 0 \quad (2.18b)$$

$$\phi_M = 2\pi - \Delta\phi \quad (2.18c)$$

$$\phi_m = \phi_{m-1} + \Delta\phi \quad (2.18d)$$

ϕ_m est une suite de $M+1$ éléments.

Une DOA est alors caractérisée par le couple (θ_n, ϕ_m) . Une manière simple de représenter le jeu de DOAs obtenues selon cette méthode est d'afficher ces points sur une sphère unité. On la visualise alors selon l'axe principal d'arrivée du faisceau, l'axe Z , i.e. on projette la sphère dans le plan (X,Y) . De cette manière, les DOAs sont les points d'intersection des parallèles et des méridiens représentés sur la Figure 2.16. Dans ce cas, on a pris $N = 9$ et $M = 19$.

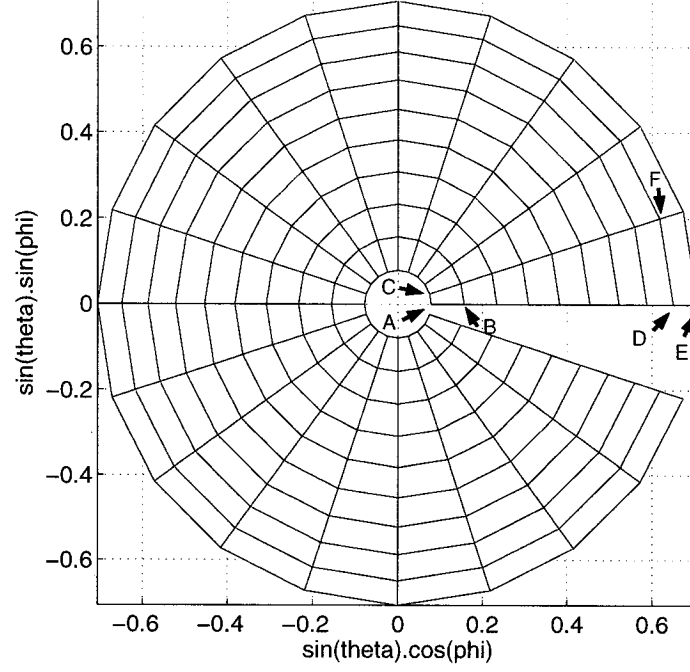


FIG. 2.16 *Maillage arithmétique en θ et ϕ*

Le défaut majeur de cette méthode réside dans sa dissymétrie : considérons les trois points A , B et C définis respectivement par (θ_0, ϕ_0) , (θ_1, ϕ_0) et (θ_0, ϕ_1) . On remarque que $AB > AC$. On appelle AB la distance angulaire sur un arc de grand cercle passant par les points A et B , où un grand cercle est défini comme un cercle de rayon unitaire dont le centre est situé au centre de la sphère. Cette définition est valable également pour les autres distances utilisées dans cette section. Considérons maintenant trois nouveaux points D , E et F définis respectivement par (θ_8, ϕ_0) , (θ_9, ϕ_0) et (θ_8, ϕ_1) . De même, on peut voir que $DE = AB < DF$ (le fait que DE apparaisse plus courte que AB est dû à la projection de cette portion de sphère sur le plan (X, Y)).

Le problème est donc la non conservation des distances AC et DF . En effet, à nombre de points fixé pour l'entraînement de MLP_x, cette méthode de sélection

des DOAs n'est pas optimale puisque les erreurs (pendant la phase de validation) seront plus grandes sur le bord de la calotte qu'en son centre. En effet, l'erreur d'estimation pour une DOA quelconque au point $G(\theta, \phi)$ dépend de son éloignement sur la calotte avec le point $I(\theta_i, \phi_j)$ (I est donc un point du maillage) défini de telle façon à ce que la distance GI soit minimale. Ces nouveaux points, ainsi que ceux qui sont introduits dans le prochain paragraphe, sont placés sur la Figure 2.17. Pour une meilleure visibilité, on a pris un quadrillage moins serré.

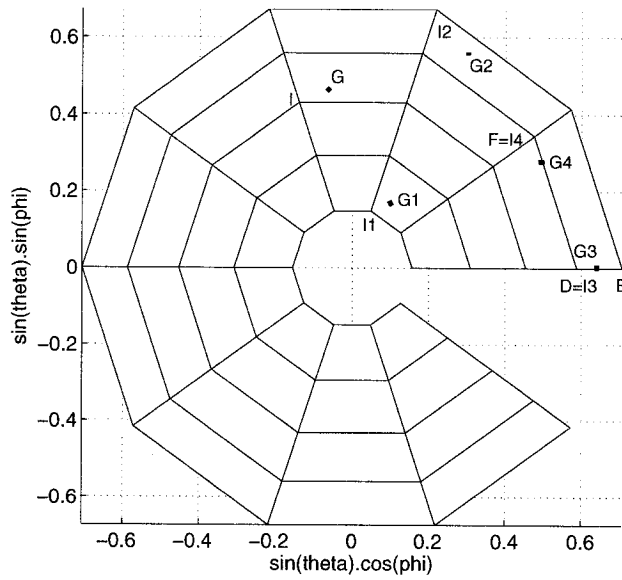


FIG. 2.17 Maillage arithmétique simplifié

Soit G_1 un point près du pôle, I_1 son plus proche voisin, G_2 un point près du bord extérieur de la calotte et I_2 son plus proche voisin. On peut supposer que :

$$P[G_2 I_2 > G_1 I_1] > 0.5 \quad (2.19)$$

Soit $G_3 \in [DE]$ (on considère le segment $[DE]$ comme un arc de cercle), I_3 son

plus proche voisin ($I_3 = D$ ou E), $G_4 \in [DF]$ et I_4 son plus proche voisin ($I_4 = D$ ou F).

$$P[G_4 I_4 > G_3 I_3] > 0.5 \quad (2.20)$$

Les équations 2.19 et 2.20 (qui sont des inégalités strictes et non des égalités) prouvent que ce maillage n'est pas le meilleur. La prochaine partie explique la méthode pour générer un maillage optimal.

2.5.1.2 Maillage optimal

La solution consiste à considérer un maillage carré et à l'appliquer sur une sphère de telle manière à créer des DOAs équiréparties sur le cône d'intérêt. On raisonne cette fois en coordonnées cartésiennes pour commencer. On définit un carré de longueur deux que l'on quadrille et l'on en extrait le cercle inscrit quadrillé comme représenté dans le schéma de gauche de la Figure 2.18.

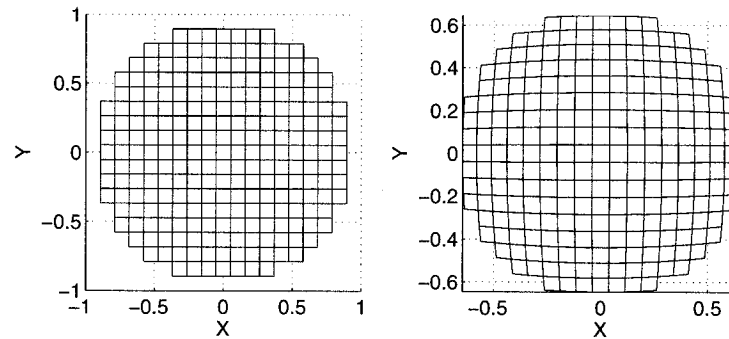


FIG. 2.18 *Maillage carré d'un disque de rayon unité à gauche, compression du maillage du disque à droite*

On transforme le disque quadrillé unitaire en un disque quadrillé de rayon $\frac{\pi}{4}$ (ce facteur $\frac{\pi}{4}$ correspond en fait au demi-angle du cône d'intérêt). On appelle alors r la distance séparant chaque intersection du quadrillage avec le centre du nouveau disque. On veut que ce maillage représente la projection sur le plan (X,Y) des DOAs ponctuant la sphère unité. Pour que la densité de DOAs sur la sphère soit équirépartie, il faut compresser le maillage sur le bord du disque pour tenir compte de l'effet de resserrement des parallèles quand on regarde un globe selon l'axe des pôles par exemple. Le facteur de compression est en $\frac{\sin r}{r}$. Le schéma de droite de la Figure 2.18 représente le disque ainsi modifié. On peut résumer mathématiquement l'ensemble de ces transformations par les équations qui vont suivre. On désigne par x_{ij} et y_{ij} les coordonnées du point d'intersection de la i^{eme} ligne avec la j^{eme} colonne. On suppose que l'on quadrille le carré en $N + 1$ lignes et colonnes.

$$x_{ij} = -1 + \frac{2(j-1)}{N} \quad (2.21a)$$

$$y_{ij} = 1 - \frac{2(i-1)}{N} \quad (2.21b)$$

Le disque \mathcal{D} de rayon unité est défini par l'équation 2.22 :

$$(x_{ij}, y_{ij}) \in \mathcal{D} \Leftrightarrow x_{ij}^2 + y_{ij}^2 \leq 1 \quad (2.22)$$

On désigne maintenant par x_{ij}^c et y_{ij}^c les coordonnées des points d'intersection du maillage du disque modifié représenté sur le schéma de droite de la Figure 2.18.

$$r_{ij} = \frac{\pi}{4} \sqrt{x_{ij}^2 + y_{ij}^2} \quad (2.23a)$$

$$x_{ij}^c = \frac{\pi}{4} x_{ij} \frac{\sin r_{ij}}{r_{ij}} \quad (2.23b)$$

$$y_{ij}^c = \frac{\pi}{4} y_{ij} \frac{\sin r_{ij}}{r_{ij}} \quad (2.23c)$$

La Figure 2.19 permet de visualiser en 3D les DOAs obtenues sur une sphère unité.

Les DOAs sont toujours aux points de jonction du quadrillage.

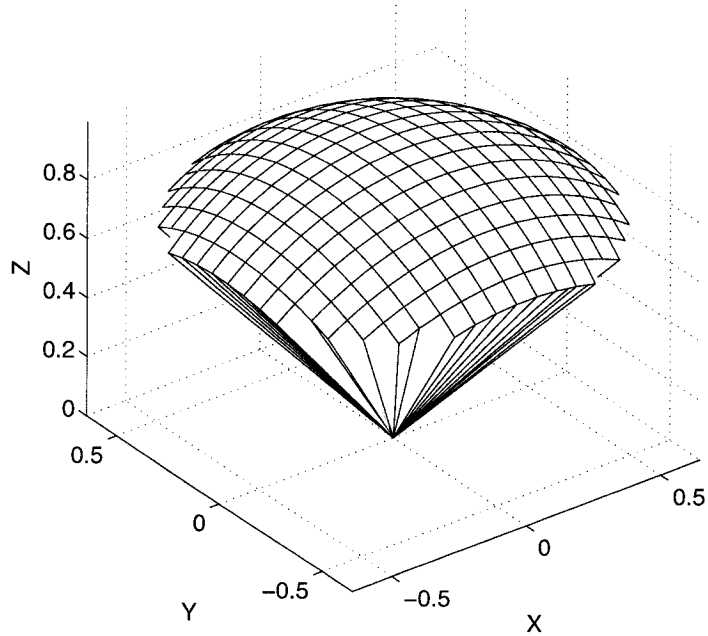


FIG. 2.19 *Maillage optimal des DOAs sur une calotte sphérique de rayon unité*

On remarque que cette fois, les inégalités 2.19 et 2.20 deviennent des égalités dans ce cas. On a donc bien atteint nos objectifs pour concevoir un maillage optimal de DOAs. On va générer ensuite les signaux de sortie des combineurs pour des DOAs réparties selon ce type de maillage lors des phases d'entraînement et de validation des MLPs.

2.5.2 Entraînement de MLP_x (ou MLP_y)

Maintenant que le principe d'entraînement d'un MLP est connu, on va présenter plus précisément ceux utilisés pour MLP_x, MLP_y et MLP_d. Considérons MLP_x

dans un premier temps (ou MLPy). On verra après le cas de MLPd. On génère un maillage optimal du cône d'ouverture 90° de façon à obtenir les huit signaux $(X_{1x,n}, \dots, X_{8x,n})$ pour chacune de ces DOAs, dont l'angle θ_x est connu. On a ainsi des couples (entrée, sortie) déterminés pour entraîner MLPx. En notant $(\vec{X}_{x,n} = X_{1x,n}, \dots, X_{8x,n})$, les couples (entrée, sortie) sont $(\vec{X}_{x,n}, \theta_x)$. On suppose que l'on a généré N échantillons, i.e. N couples $(\vec{X}_{x,n}, \theta_x)$. On applique alors l'algorithme de rétropropagation d'erreurs décrit plus haut pour optimiser nos poids synaptiques. La fonction coût est ici différente de l'erreur quadratique moyenne. On utilise plutôt la fonction J_{sin} définie par l'équation 2.24 :

$$J_{sin} = \frac{1}{N} \sum_{i=1}^N \sin\left(\frac{y_i - t_i}{2}\right)^2 \quad (2.24)$$

avec y_i la sortie de MLPx généré avec le i^{eme} échantillon et t_i l'angle θ_x correspondant à cet échantillon. De cette façon, la fonction erreur s'annule en 0 et 2π , ce qui tient compte de la 2π -périodicité du problème. Elle est maximale et égale à 1 en π .

Pour choisir la structure neuronale optimale, on se fixe un certain nombre de neurones dans la couche cachée. On entraîne alors MLPx sur un nombre croissant d'époques. Soit N_e le nombre d'époques d'entraînement. On utilise alors des échantillons de validation différents des échantillons d'entraînement pour calculer l'erreur d'estimation à chaque époque d'entraînement. Si on a pris comme points d'entraînement ceux définis par un maillage optimal, les points de validation seront ceux situés au milieu des cases du quadrillage de la Figure 2.19. On utilise donc pratiquement autant de points de validation que de points d'entraînement : si on utilise 10000 points d'entraînement (maillage 100×100), on aura 9801 points de validation (maillage 99×99).

Il existe toujours un nombre d'époques optimal, noté N_e^* , qui donnera une erreur de validation minimale. En entraînant indéfiniment un MLP, on pourrait annuler

complètement l'erreur calculée sur les échantillons d'entraînement. Mais ceci ne veut pas dire pour autant que notre réseau a été bien entraîné. Les données des échantillons sont bruitées et annuler l'erreur d'entraînement revient en fait à apprendre le bruit de mesure, plutôt que de découvrir la distribution sous-jacente. C'est le fameux phénomène de surapprentissage (Haykin, 1999). Pour l'éviter, on utilise des échantillons de validation dont on veut minimiser l'erreur, que l'on appelle "erreur de validation" E_v (différente de "l'erreur d'entraînement" E_t). La courbe $E_v(N_e)$ passe par un minimum en N_e^* . Pour un nombre fixé de neurones dans sa couche cachée, MLPx est alors entraîné de façon optimale.

Il reste ensuite à déterminer le nombre optimal de neurones dans la couche cachée. On trouve la configuration optimale par dichotomie : on se fixe une plage de valeurs possibles dans laquelle se trouve le nombre optimal de neurones dans la couche cachée, N_n^* . On prend par exemple la plage $[1;60]$. On calcule $E_v(N_e^*)$ pour des MLPs à 1, 10, 20, 30, 40, 50 et 60 neurones dans la couche cachée. De proche en proche, on peut remonter à la structure optimale tant du point de vue du nombre de neurones que du nombre d'époques d'entraînement.

2.5.3 Entraînement de MLPd

Le principe reste le même pour entraîner MLPd. On utilise le même algorithme de rétropropagation d'erreur en utilisant la fonction "trainlm". La première différence se fait au niveau de la fonction de transfert du neurone de sortie, qui est en seuil, au lieu d'être linéaire, et pour cause : la sortie vaut 0 si le signal d'arrivée n'appartient pas au cône d'intérêt, et 1 sinon. Si c'est le cas, MLPx et MLPy sont activés pour estimer précisément la DOA. La fonction coût est cette fois l'erreur quadratique

moyenne J_{mse} définie par :

$$J_{mse} = \frac{1}{2N} \sum_{i=1}^N (y_i - t_i)^2 \quad (2.25)$$

Les couples (entrée, sortie) sont nécessairement différents de ceux utilisés pour entraîner MLPx ou MLPy. En effet, cet étage permet de distinguer les DOAs contenues ou non dans le cône d'intérêt. La DOA appartient par hypothèse au demi-espace $z > 0$. On doit donc échantillonner ce demi-espace, i.e. le cône d'ouverture 180° et d'axe Z . On considère comme précédemment une calotte sphérique, qui est cette fois une demi-sphère, et on l'échantillonne de façon optimale comme représenté dans la Figure 2.20.

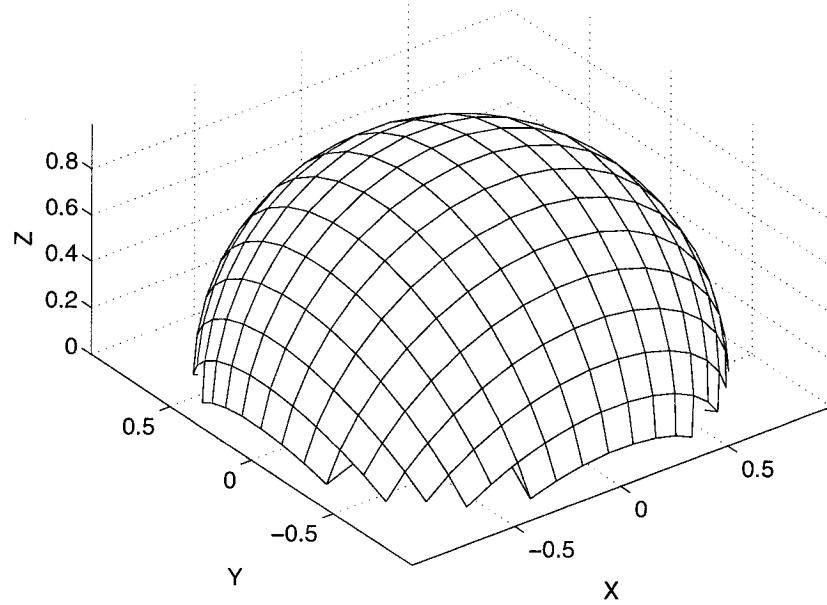


FIG. 2.20 *Maillage optimal pour entraîner MLPd*

On va générer les 16 signaux de puissance normalisés en sortie des deux combineurs correspondant à ces DOAs. Le vecteur $\vec{X}_n = (X_{1x,n}, \dots, X_{8x,n}, X_{1y,n}, \dots, X_{8y,n})$ forme le vecteur d'entrée de MLPd. On optimise la structure neuronale du point de vue nombre d'époques d'entraînement dans un premier temps, puis on choisit la couche cachée offrant les meilleures performances, i.e. celle qui minimise J_{mse} . On choisit arbitrairement une plage de variations $[1;60]$ pour le nombre de neurones dans la couche cachée. On fixe cette borne supérieure à 60 pour des raisons pratiques : étant donné le nombre de points d'entraînements, 30000 échantillons souvent, l'ordinateur à disposition (1 Go de RAM), ne peut dépasser physiquement cette limite. Heureusement, il s'avère qu'en pratique, la couche optimale comporte moins de neurones que cette borne.

2.5.4 Ajout du bruit et traitements pour en réduire son effet

En pratique, du bruit perturbe l'estimation de la DOA : bruit de réception au niveau des antennes, bruit dans les lignes de transmission, bruit au niveau de l'électronique dans le récepteur RF surtout. Des effets de distorsion du signal apparaissent aussi et sont difficiles à modéliser. Dans le cadre de nos simulations, on rajoute un bruit gaussien de rapport signal à bruit (RSB) fixé. On a le choix, pour générer nos points d'entraînement, de rajouter du bruit ou non. Par contre, dans la phase de validation, le bruit est omniprésent puisque l'on veut se rapprocher le plus près possible de la réalité. Dans la phase d'entraînement aussi il est judicieux de rajouter du bruit. En effet, dans un problème de classification par exemple, l'hyperplan optimal (une droite dans un problème à 2D) sera trouvé plus facilement si l'on utilise des points bruités. Cette caractéristique est dépeinte dans la Figure 2.21.

La figure de gauche représente des couples (entrée, sortie) de points non bruités. En entraînant un RNA avec ces points, on n'arrivera pas à trouver la droite de

séparation optimale, i.e. celle maximisant la marge entre les deux droites de support, d'équations $y = 0.6$ et $y = 0.4$. Cet hyperplan optimal est donc la droite d'équation $y = 0.5$. Par contre, en utilisant des couples (entrée, sortie) de points bruités comme sur la figure de droite (ajout d'un bruit gaussien suivant une loi gaussienne de moyenne nulle et d'écart type $1/5$), l'entraînement du RNA conduira à l'hyperplan optimal comme droite de séparation des deux classes.

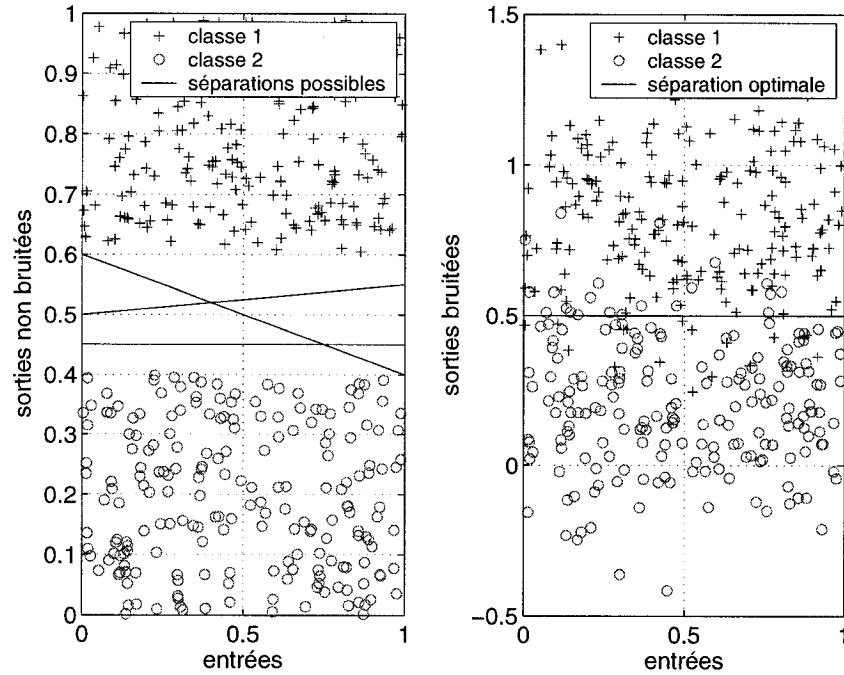


FIG. 2.21 Intérêt du bruit dans la phase d'entraînement d'un problème de classification

Une autre manière consisterait à utiliser des RNAs de type *SVM* qui maximisent la marge, mais cette méthode serait moins rapide à mettre en œuvre que de rajouter du bruit sur nos données d'entraînement. Le prochain paragraphe expose comment on a introduit du bruit dans notre modèle.

2.5.4.1 Ajout du bruit de RSB fixé

On ajoute un bruit complexe sur les signaux complexes de sortie des deux combineurs. Dans notre modèle, cette opération se fait juste avant celle consistant à prendre la puissance de ces signaux. On peut dire que le bruit est gaussien dans la mesure où l'amplitude du bruit est une variable aléatoire qui suit une distribution gaussienne $N(0, \sigma^2)$. La variance de la loi gaussienne est fixée par le RSB donné en hypothèse. La phase du bruit est elle une variable aléatoire décrivant une loi uniforme sur la plage $[-\pi, \pi]$. Les bruits $n_{ix}, i = 1 \dots 8$ (respectivement $n_{iy}, i = 1, \dots, 8$) sont rajoutés aux signaux $x_{ix}, i = 1, \dots, 8$ (respectivement $x_{iy}, i = 1, \dots, 8$) donnés dans les équations 2.26.

$$x_{1x} = -j \frac{A_{x1} - A_{x2}}{2} \quad (2.26a)$$

$$x_{2x} = -\frac{A_{x1} + A_{x2}}{2} \quad (2.26b)$$

$$x_{3x} = -\frac{jA_{x2} + A_{x3}}{2} \quad (2.26c)$$

$$x_{4x} = -\frac{A_{x2} + jA_{x3}}{2} \quad (2.26d)$$

$$x_{5x} = -j \frac{A_{x1} - A_{x3}}{\sqrt{8}} \quad (2.26e)$$

$$x_{6x} = -\frac{A_{x1} + A_{x3}}{\sqrt{8}} \quad (2.26f)$$

$$x_{7x} = -\frac{jA_{x1} + A_{x3}}{\sqrt{8}} \quad (2.26g)$$

$$x_{8x} = -\frac{A_{x1} + jA_{x3}}{\sqrt{8}} \quad (2.26h)$$

Le bruit $n_{ix}, i = 1, \dots, 8$ est défini selon le modèle de l'équation 2.27.

$$n_{ix} = a_{ix} e^{j\alpha_{ix}} \quad (2.27)$$

avec a_{ix} une variable aléatoire gaussienne $\mathcal{N}(0, \sigma^2)$ et α_{ix} une variable aléatoire uniforme sur $[-\pi, \pi]$.

La variance σ^2 est calculée à partir du RSB. Il faut pour cela connaître la puissance maximale parmi les signaux $X_{ix}, i = 1, \dots, 8$ (et $X_{iy}, i = 1, \dots, 8$) définis par les équations 2.8. Soit X_{max} cette valeur :

$$X_{max} = \max(\max_{i=1, \dots, 8} X_{ix}, \max_{i=1, \dots, 8} X_{iy}) \quad (2.28)$$

La définition du RSB est donnée en équation 2.29.

$$RSB = 10 \log_{10} \left(\frac{X_{max}}{\sigma^2} \right) \quad (2.29)$$

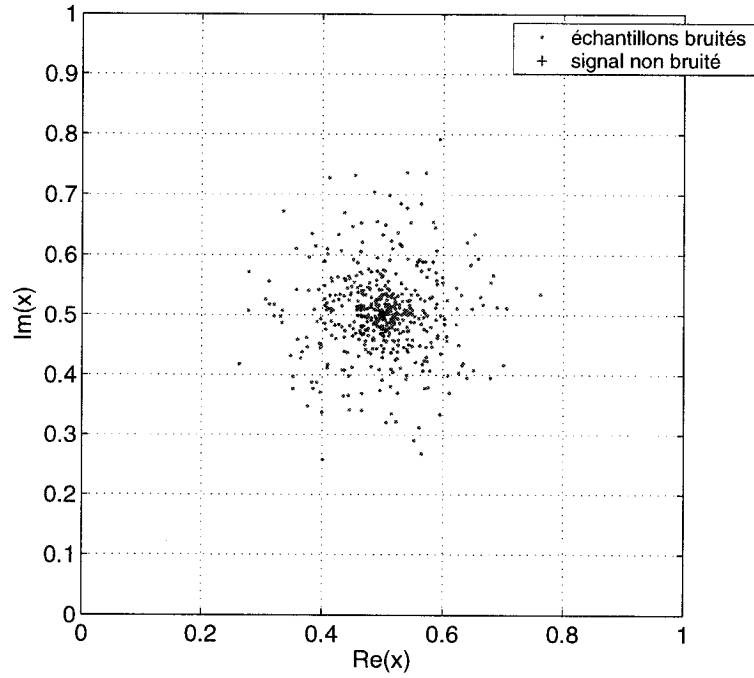


FIG. 2.22 *Comparaison des signaux bruités avec le signal originel*

En-se fixant un RSB de 20 dB et en supposant par exemple $X_{max} = 1$, on obtient $\sigma^2 = 0.01$. La Figure 2.22 trace 500 signaux bruités pour un signal $x_{ix} = 0.5 + 0.5j$ en respectant ces hypothèses.

Le bruit n_{ix} (n_{iy}) est ensuite additionné au signal x_{ix} (x_{iy}). On en prend la norme au carré pour avoir les signaux de puissance bruités X_{ix}^b en sortie du combineur :

$$X_{ix}^b = \|x_{ix} + n_{ix}\|^2, i = 1, \dots, 8 \quad (2.30)$$

On normalise ensuite ces signaux, on reprend l'équation 2.9a pour obtenir la nouvelle équation 2.31.

$$X_{ix,n}^b = \frac{X_{ix}^b}{\max_{j=1,\dots,8} X_{jx}^b}, i = 1, \dots, 8 \quad (2.31)$$

Ce sont ces signaux bruités, normalisés et numérisés qui servent à entraîner et valider les MLPs de notre système.

2.5.4.2 Atténuation de l'effet du bruit sur l'estimation de la DOA par pré- et post-traitement, selon (Coudyser, 2003)

Il est possible de réduire l'effet du bruit sur la qualité de l'estimation en appliquant un pré-traitement et/ou un post-traitement qui consistent à prendre des valeurs moyennées susceptibles d'être meilleures. Cette opération de filtrage est efficace parce que le bruit additif est de moyenne nulle et donc, qu'il n'introduit pas de biais sur l'estimation de la DOA. Il faut garder à l'esprit que notre système fonctionne en temps réel et qu'il fournit en continu des signaux de puissance ainsi que des estimations de la DOA. Ces opérations ne se font pas au prix du débit de données dès que le système fonctionne en continu, ce qui est le cas pour une utilisation future.

Dans le cas du pré-traitement, on prend une valeur moyenne $\overline{X_{ix,n}^b}$ de chaque signal de puissance $X_{ix,n}^b$ en sortie des combineurs sur N_{pre} valeurs successives et MLPx fait alors son travail à partir de ces signaux moyennés (idem pour MLPy).

$$\overline{X_{ix,n}^b} = \frac{1}{N_{pre}} \sum_{j=1}^{N_{pre}} X_{ix,n}^b(j), \quad i = 1, \dots, 8 \quad (2.32)$$

Pour le post-traitement, l'opération de filtrage est effectuée sur les valeurs de sortie de MLPx, i.e. sur les estimations $\tilde{\theta}_x$ de θ_x (idem pour MLPy). Soit $\bar{\theta}_x$ cette valeur moyenne réalisée sur N_{post} valeurs successives.

$$\bar{\theta}_x = \frac{1}{N_{post}} \sum_{j=1}^{N_{post}} \tilde{\theta}_x(j) \quad (2.33)$$

Il est tout-à-fait possible de combiner les effets bénéfiques du pré-traitement et du post-traitement pour réduire au maximum les effets du bruit. Ceux-ci sont rapportés dans la prochaine sous-section.

2.5.5 Résultats d'estimation avec du bruit et sans réflexion parasite avec MLPx et MLPy

La méthode conduisant à l'élaboration du système optimal d'estimation (travail sur MLPx et MLPy) est légèrement différente de celle exposée précédemment. En effet, cette dernière est fastidieuse et prend du temps. On a plutôt intérêt à adopter celle-ci, légèrement différente, donnant des résultats similaires et surtout beaucoup plus rapide à mettre en œuvre.

Voici les étapes majeures de l'algorithme :

- on se donne un quadrillage initial carré de 200×200 points

- on le modifie pour obtenir $10000\pi \approx 31400$ DOAs équiréparties sur le cône d'intérêt d'ouverture 90° .
- on génère les 16 signaux de puissance bruités $X_{ix,n}^b$ et $X_{iy,n}^b$ (sans pré-traitement ni post-traitement dans un premier temps) dont Le RSB est fixé à 20 dB par rapport au signal X_{max} . La distance émetteur-récepteur d_1 n'a pas d'importance ici. On la prend égale à $\sqrt{2} m$. Le récepteur est lui situé à une hauteur $h = 1 m$ d'un sol parfaitement absorbant (pas de réflexion parasite générée donc). La fréquence de travail est fixée à 5.8 GHz. Le gain des antennes est pris en $\cos \theta$.
- on se fixe dans un premier temps un nombre d'époques d'entraînement pour MLPx et MLPy. On le prend égal à 30 : en général, la phase d'apprentissage des RNAs a déjà convergé au bout de 30 époques, étant donné le nombre de points d'entraînement et la taille du réseau.
- on entraîne MLPx et MLPy avec la fonction "trainlm" selon la fonction coût J_{sin} . On teste les structures ayant entre 5 et 60 neurones, de 5 en 5, soit 12 configurations possible pour MLPx et MLPy. On valide ensuite les résultats en testant le réseau entraîné sur un jeu de DOAs différent de celui de la phase d'entraînement. Pour cela, on utilise un quadrillage similaire au premier, mais décalé : les intersections du quadrillage de validation sont placés aux centres des carreaux formant le quadrillage d'entraînement. De cette façon, les DOAs de validation sont le plus éloignées possible des DOAs utilisées pendant la phase d'apprentissage du MLP.
- une fois choisi le nombre de neurones dans la couche cachée pour MLPx et MLPy, on essaie d'optimiser le nombre d'époques d'entraînement cette fois. On procède de même, on choisira l'entraînement qui minimisera l'erreur angulaire moyenne sur les DOAs de validation (au nombre de N_{val}). L'erreur angulaire moyenne ϵ_{av} est définie par l'équation 2.34.
- La structure neuronale optimale est alors adoptée. On peut ensuite étudier sa robustesse au bruit et son atténuation par filtrage.

$$\epsilon_{av,x} = \frac{1}{N_{val}} \sum_{j=1}^{N_{val}} |\tilde{\theta}_x(j) - \theta_x(j)| \quad (2.34a)$$

$$\epsilon_{av,y} = \frac{1}{N_{val}} \sum_{j=1}^{N_{val}} |\tilde{\theta}_y(j) - \theta_y(j)| \quad (2.34b)$$

Une remarque importante à faire concerne le choix du nombre N_{ent} de points d'entraînement par rapport au nombre total N_w de poids synaptiques utilisés dans le MLP. La règle du pouce classique conseille de vérifier la relation suivante :

$$N_{ent} > \frac{N_w}{\epsilon_d} \quad (2.35)$$

ϵ_d est l'erreur que l'on veut effectivement atteindre. Dans notre cas, on peut se fixer une erreur de l'ordre de 0.1° , soit $\epsilon_d = 0.1$. Pour le calcul, on considère la taille maximale des MLPs que l'on entraîne, soit des MLPs comportant 60 neurones sur leur couche cachée. Avec des vecteurs d'entrée de huit composantes et une couche de sortie composée d'un seul neurone, on arrive à $N_w = 540$. On en déduit qu'il est nécessaire de prendre plus de 5400 points d'entraînement. Cette borne inférieure est largement dépassée avec nos quelques 30000 points d'entraînement.

La Figure 2.23 représente le travail d'optimisation effectué sur MLPx et MLPy. La courbe de gauche représente l'erreur angulaire moyenne $\epsilon_{av,x}$ et $\epsilon_{av,y}$ en fonction du nombre de neurones dans la couche cachée de MLPx et MLPy, en se fixant un apprentissage sur 30 époques dans tous les cas. A précision identique, on préférera prendre un MLP ayant moins de neurones : la régression de fonction effectuée par le MLP est plus simple, mais aussi plus robuste. On adopte finalement des structures à 30 neurones dans la couche cachée pour MLPx et MLPy. La courbe de droite trace l'erreur angulaire moyenne $\epsilon_{av,x}$ et $\epsilon_{av,y}$ en fonction du nombre d'époques

d'entraînement cette fois, en se fixant des structures à 30 neurones dans la couche cachée. De même, à performance égale, on choisira l'apprentissage le plus rapide, celui qui en fin de compte approche le mieux la distribution sous-jacente, sans "apprendre" le bruit. 30 époques d'entraînement paraît être un bon compromis.

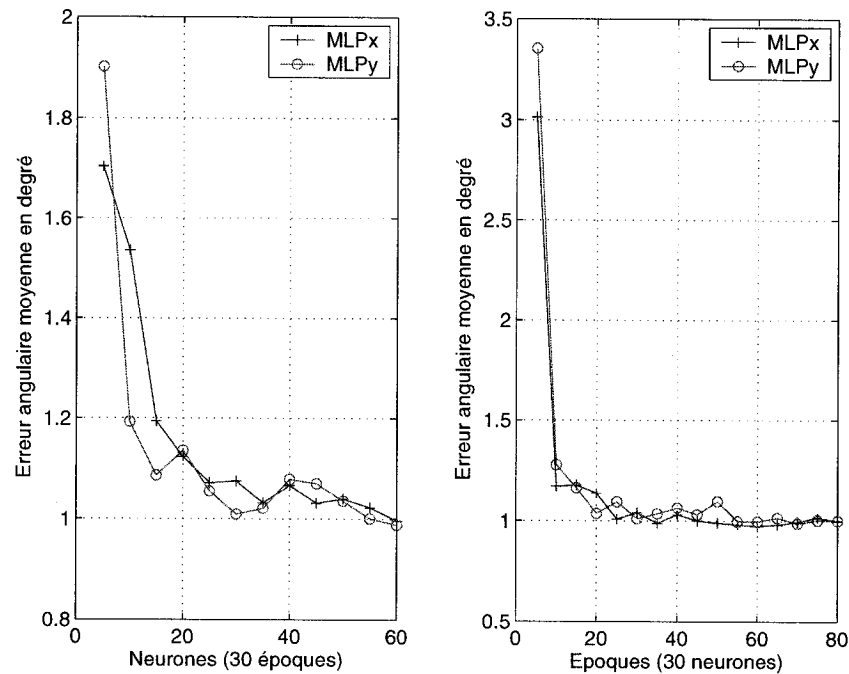


FIG. 2.23 *Optimisation du nombre de neurones dans la couche cachée (sur 30 époques) à gauche, et du nombre d'époques d'entraînement (avec 30 neurones) à droite, pour MLPx et MLPy*

Conclusion : on entraîne MLPx et MLPy, des réseaux à 30 neurones dans la couche cachée, sur 30 époques, dans le cadre de l'entraînement sans réflexion parasite.

Les prochains paragraphes exposent les résultats que l'on obtient avec la configuration neuronale optimisée. La Figure 2.24 montre les différentes estimations faites par MLPx et MLPy en fonction des angles d'arrivée réels θ_x et θ_y . Plus de 30000 points de validation y sont représentés, d'où un certain manque de visibilité.

Pour apprécier la qualité de l'estimation, on a représenté les points par un jeu de données de validation obtenues en considérant des tranches angulaires de 1° . Toutes les estimations faites dans la même tranche sont remplacées par une seule valeur obtenue en prenant la moyenne de toutes les estimations de la tranche. Il en résulte une visualisation aérée mais aussi une précision artificiellement améliorée par l'annulation des erreurs positive et négative qui se compensent mutuellement.

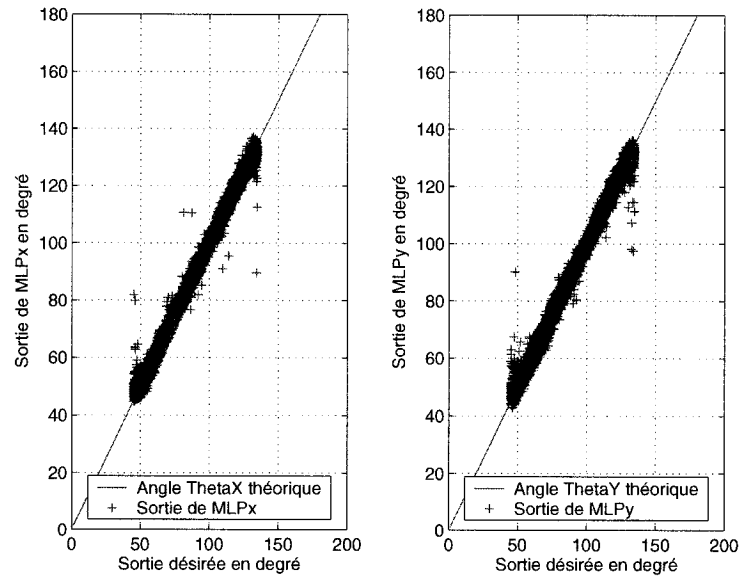


FIG. 2.24 Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite

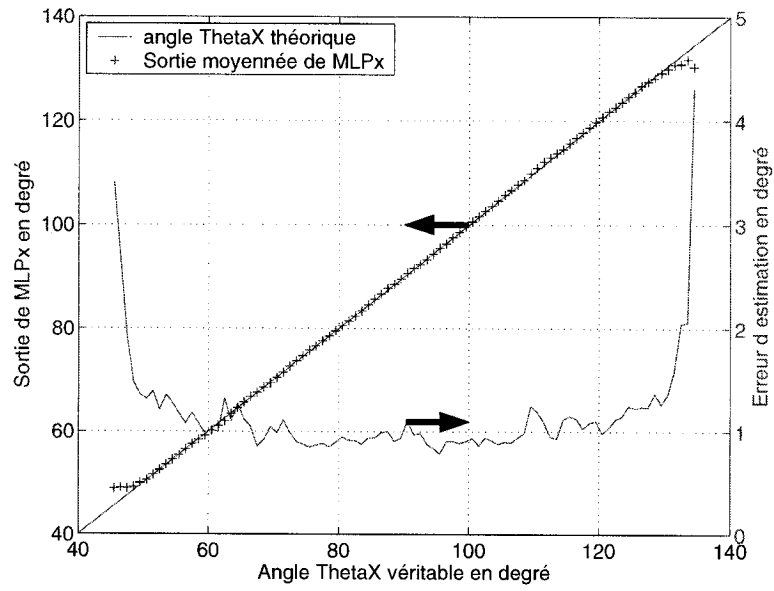


FIG. 2.25 Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx

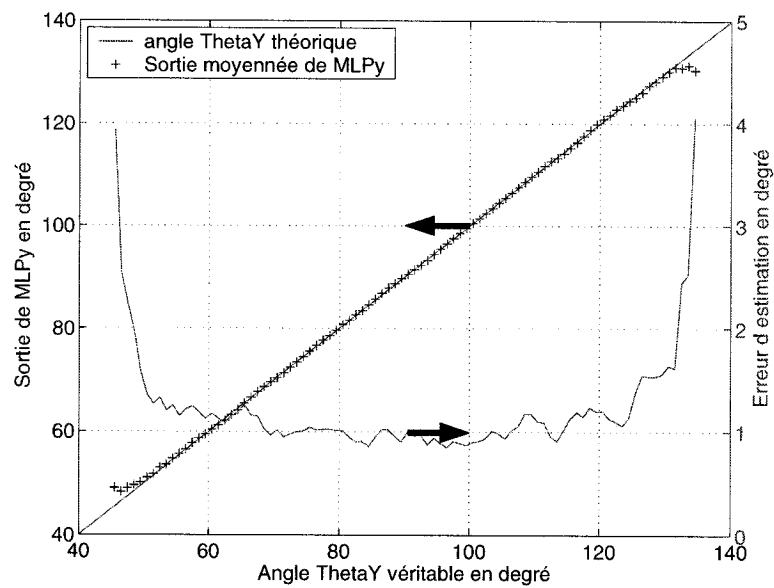


FIG. 2.26 Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy

On trace également sur le même graphe l'erreur absolue d'estimation $\bar{\epsilon}_x$ (ou $\bar{\epsilon}_y$) sur chaque tranche de 1° en se basant sur l'équation 2.36.

$$\bar{\epsilon}_x = \frac{1}{N_{abs}} \sum_{j=1}^{N_{abs}} |\tilde{\theta}_x(j) - \theta_x(j)| \quad (2.36)$$

N_{abs} est le nombre de points compris dans la tranche de 1° considérée. Ces deux courbes sont tracées pour MLPx d'une part (Figure 2.25) et MLPy d'autre part (Figure 2.26). On peut remarquer un léger effet de décrochage de l'estimation sur les bords du cône d'intérêt. Ce n'est pas très grave puisqu'une bonne précision est surtout requise pour les DOAs correspondant à la direction principale d'arrivée, celle selon l'axe Z . Pour tenir compte de cette caractéristique, on peut introduire une fonction coût pondérant différemment une erreur de la DOA selon qu'elle est proche ou non de la DOA principale. On va définir en fait trois erreurs moyennes différentes pour apprécier la qualité d'estimation des MLPs : l'erreur angulaire moyenne (en valeur absolue) ϵ_{av} , l'erreur quadratique moyenne ϵ_{ms} et le coût angulaire moyen ϵ_{mc} . Ces trois valeurs sont données en degré et par les équations 2.37.

$$\cos(e_{3D}(j)) = D\hat{O}A(j) \cdot D\hat{O}A(j) \quad (2.37a)$$

$$\epsilon_{av} = \frac{1}{N_{val}} \sum_{j=1}^{N_{val}} |e_{3D}(j)| \quad (2.37b)$$

$$\epsilon_{ms} = \sqrt{\frac{1}{N_{val}} \sum_{j=1}^{N_{val}} |e_{3D}(j)|^2} \quad (2.37c)$$

$$\epsilon_{mc} = \frac{1}{N_{val}} \sum_{j=1}^{N_{val}} |f_w(j)e_{3D}(j)| \quad (2.37d)$$

On définit avant l'erreur d'estimation $e_{3D}(j)$ pour la j^{ieme} DOA (en degré) parmi N_{val} points : son cosinus est égal au produit scalaire entre les deux vecteurs unitaires

définissant la DOA réelle et la DOA estimée ($D\hat{O}A(j)$ et $D\hat{\hat{O}}A(j)$ respectivement). La fonction $f_w(j)$ est une fonction de pondération qui augmente le poids des erreurs faites le long de l'axe Z et réduit l'importance de celles faites près du bord du cône. Elle dépend donc de l'angle θ . Par défaut, on a pris la fonction linéaire $f_w(\theta)$ donnée par l'équation 2.38.

$$f_w(\theta) = 2 - \frac{\theta}{30} \quad (2.38)$$

$f_w(\theta = 0^\circ) = 2$ et $f_w(\theta = 45^\circ) = 0.5$.

Le Tableau 2.1 expose les résultats obtenus selon les différentes définitions d'erreurs, sous un niveau de bruit de -20 dB et montre l'effet du pré-traitement : la précision est nettement améliorée.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}
RSB = 20 dB et $N_{pre} = 1$	1.04°	1.10°	1.72°	2.15°	1.60°
RSB = 20 dB et $N_{pre} = 30$	0.25°	0.33°	0.46°	0.53°	0.44°

TAB. 2.1 *Comparaison des résultats effectués sur des signaux bruités (RSB = 20 dB) avec et sans pré-traitement*

Conclusion : il est possible d'estimer la direction d'arrivée d'un faisceau frappant notre système selon la méthode exposée dans ce chapitre, avec une bonne précision et ce malgré un niveau de bruit important (RSB = 20 dB). La prochaine étape de notre étude consiste à tester notre système quand une réflexion parasite générée par une route cause des interférences avec le signal principal et perturbe les mesures. On va montrer de quelle manière on peut en partie s'affranchir des influences de cette dernière.

CHAPITRE 3

EFFETS D'UNE RÉFLEXION PARASITE ET OPTIMISATION DE LA STRUCTURE DU SYSTÈME

Dans ce chapitre, on modélise l'effet de la surface d'une route du point de vue électromagnétique. L'interface air-asphalte génère une réflexion parasite du signal principal émis par la voiture de tête dans le cas d'une poursuite automobile. La réception au niveau du véhicule de queue en est perturbée. Nous allons voir dans quelle mesure et comment on peut atténuer l'effet de cette réflexion sur l'estimation de la direction d'arrivée.

3.1 Modélisation d'une route et de la réflexion par l'optique géométrique

Pour simplifier, on considère que les deux véhicules de notre application évoluent sur une route homogène modélisée par un plan infini. En pratique, selon l'épaisseur des différentes strates de matériaux constituant la route, différentes réflexions peuvent se produire. On admettra que l'épaisseur de la couche de matériau à la surface est telle qu'elle n'autorise qu'une seule réflexion. On réduit nos véhicules à de simples points, le premier étant l'émetteur E , le deuxième le récepteur R . La Figure 3.1 schématise le modèle adopté.

Afin de vérifier si notre système de détection est performant dans tout le cône d'intérêt, l'émetteur E est positionné sur la portion de sphère de centre R et de rayon d constant. Le récepteur R est fixe et situé à une hauteur h du sol. Le sol génère un signal réfléchi selon les lois de l'optique géométrique. Le point d'incidence I varie avec la position de E . L'angle ψ est lui aussi fonction de la position de E .

On a représenté également l'angle θ_y sur le schéma.

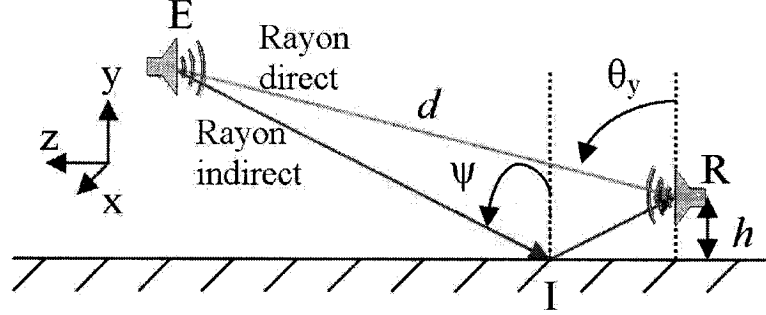


FIG. 3.1 *Modèle de réflexion parasite utilisé dans nos simulations*

On utilise la théorie de l'optique géométrique pour modéliser les faisceaux par des rayons rectilignes. La réflexion est abordée de cette manière dans un premier temps, avant d'en voir les caractéristiques électromagnétiques dans la suite.

Dans les prochains paragraphes, on donnera les coordonnées des points utiles pour notre problème dans le repère cartésien de centre R . On pourra alors en déduire les angles et les distances utiles à la résolution de notre problème.

Le point R est au centre du repère et a pour coordonnées $(0, 0, 0)$, le point E $(d \sin \theta \cos \phi, d \sin \theta \sin \phi, d \cos \theta)$ et le point I $(x_i, -h, z_i)$. On définit également le point R' comme étant l'image du point R par rapport au plan de la route d'équation $y = -h$. R' a pour coordonnées $(0, -2h, 0)$.

La distance parcourue par le signal direct vaut d . Celle-ci est fixée au début des simulations. On peut calculer la distance d_r parcourue par le signal réfléchi, elle est donnée dans l'équation 3.1. Elle est égale à la distance ER' .

$$d_r = \sqrt{(d \sin \theta \cos \phi)^2 + (d \sin \theta \sin \phi + 2h)^2 + (d \cos \theta)^2} \quad (3.1)$$

On a également besoin de déterminer l'angle d'incidence ψ . Il interviendra plus tard dans le calcul des coefficients de réflexion de Fresnel. On le trouve à partir de la relation vectorielle $\hat{y} \cdot \overrightarrow{ER'} = -d_r \cdot \cos \psi$. Finalement, l'angle ψ est donné par l'équation 3.3.

$$\cos \psi = \frac{2h + d \sin \theta \sin \phi}{d_r} \quad (3.2)$$

La dernière étape consiste à déterminer l'angle d'arrivée du rayon réfléchi au niveau du système de réception, au point R . On doit donc déterminer les angles des coordonnées sphériques θ_r et ϕ_r du rayon réfléchi \overrightarrow{RI} dans le repère (X, Y, Z) . D'après la Figure 3.1, on voit que :

$$\overrightarrow{RI} = \overrightarrow{RR'} + \overrightarrow{R'I} \quad (3.3a)$$

$$\overrightarrow{RI} = -2h \hat{y} + k_h \overrightarrow{R'E} \quad (3.3b)$$

$$\overrightarrow{RI} = k_h d \sin \theta \cos \phi \hat{x} + (k_h(d \sin \theta \sin \phi + 2h) - 2h) \hat{y} + k_h d \cos \theta \hat{z} \quad (3.3c)$$

k_h est le rapport de l'homothétie de centre R' transformant E en I : $\overrightarrow{R'I} = k_h \overrightarrow{R'E}$. On peut exprimer k_h facilement à partir des données du problème (équation 3.4).

$$k_h = \frac{h}{2h + d \sin \theta \sin \phi} \quad (3.4)$$

Le vecteur \overrightarrow{RI} est donc entièrement connu. On connaît aussi la distance $RI = R'I = k_h d_r$. On en déduit les angles θ_r et ϕ_r .

$$\overrightarrow{RI} \cdot \hat{z} = k_h d \cos \theta \quad (3.5a)$$

$$\overrightarrow{RI} \cdot \hat{z} = \|\overrightarrow{RI}\| \cdot \cos \theta_r = k_h d_r \cos \theta_r \quad (3.5b)$$

$$\cos \theta_r = \frac{d}{d_r} \cos \theta \quad (3.5c)$$

$$\tan \phi_r = \frac{\vec{RI} \cdot \hat{y}}{\vec{RI} \cdot \hat{x}} \quad (3.6a)$$

$$\tan \phi_r = \frac{k_h(d \sin \theta \sin \phi + 2h) - 2h}{k_h d \sin \theta \cos \phi} \quad (3.6b)$$

Quand l'émetteur E décrit la portion de sphère de centre R et de rayon d située au-dessus du sol, les angles θ et ϕ du rayon incident varient et on peut exprimer les angles θ_r et ϕ_r du rayon réfléchi en fonction de θ , ϕ , d et h . On fera varier θ entre 0 et 45°, ϕ entre 0 et 360° pour la phase d'estimation (i.e. pour l'entraînement de MLPx et MLPy). Par contre, on fera varier θ entre 0 et 90° pour la phase de détection (i.e. pour l'entraînement de MLPd). Typiquement, on a pris $h = 1 \text{ m}$ et $d = 10 \text{ m}$. La fréquence de fonctionnement du système étant de 5.8 GHz, la longueur d'onde λ est égale à 5.17 cm. Une longueur de 10 m représente ainsi environ 200λ . Nos antennes étant comprises dans un intervalle de 1.37λ , on peut appliquer l'approximation des champs lointains à notre système et considérer que l'angle d'arrivée et l'amplitude du faisceau sont les mêmes pour nos six antennes. La différence angulaire entre les antennes 1 et 3 est de l'ordre de 0.4° par exemple.

3.2 Etude de la réflexion d'un point de vue électromagnétique

3.2.1 Polarisation d'une antenne

Une antenne est polarisée elliptiquement par défaut : dans l'approximation du champ lointain, les champs électrique \vec{E} et magnétique \vec{B} émis sont orthogonaux à la direction de propagation donnée par le vecteur unitaire \hat{u}_r . Ils sont contenus dans un plan perpendiculaire à \hat{u}_r . Considérons le rayon EI de l'antenne émettrice.

L'équation 3.7 donne le vecteur \hat{u}_r correspondant.

$$\hat{u}_r = \frac{\overrightarrow{EI}}{\|\overrightarrow{EI}\|} \quad (3.7)$$

On définit deux autres vecteurs unitaires qui vont nous aider à décomposer le champ \overrightarrow{E} du rayon EI dans une base. Le premier est perpendiculaire à \hat{u}_r et au plan d'incidence (plan contenant les points E , I et R). On l'appelle \hat{u}_h . En effet, ce vecteur est toujours horizontal dans notre cas (d'où l'indice h). Le deuxième est perpendiculaire à \hat{u}_r et contenu dans le plan d'incidence. On l'appelle \hat{u}_v . Quand $\hat{u}_r = \hat{z}$, i.e. quand le rayon se propage selon la direction principale, le support de \hat{u}_v est vertical (d'où l'indice v). Enfin, la base $(\hat{u}_r, \hat{u}_h, \hat{u}_v)$ forme un trièdre direct.

On peut maintenant décomposer \overrightarrow{E} selon les composantes \hat{u}_h et \hat{u}_v comme indiqué dans l'équation 3.8

$$\overrightarrow{E} = E_h \hat{u}_h + E_v \hat{u}_v \quad (3.8)$$

Les composantes E_h et E_v sont complexes. On peut distinguer trois cas particuliers :

- si $E_h = 0$, la polarisation est linéaire verticale.
- si $E_v = 0$, la polarisation est linéaire horizontale.
- si $\frac{E_v}{E_h} = e^{\pm j\pi/2}$, la polarisation est circulaire.

On suppose que l'antenne émettrice est polarisée circulairement. On verra dans la suite quels sont les avantages et inconvénients d'une telle polarisation pour notre système. Ses composantes E_v et E_h vont être réfléchies différemment par le sol. Les composantes du faisceau réfléchi IR sont données par les coefficients de réflexion de Fresnel.

3.2.2 Coefficients de réflexion de Fresnel

Ces coefficients donnent les amplitudes et les phases des composantes horizontales et verticales du signal réfléchi, connaissant le faisceau incident. Ces coefficients sont donnés en fonction de l'angle d'incidence ψ et de des propriétés électriques du matériau réflecteur, et sont valides pour des ondes planes. On supposera que c'est le cas dans notre modèle (approximation des ondes quasi-planes). On appelle $C_h = \frac{E_{h,r}}{E_{h,d}}$ et $C_v = \frac{E_{v,r}}{E_{v,d}}$ les coefficients de réflexion de Fresnel horizontal et vertical respectivement (Les indices "d" et "r" représentent respectivement les signaux direct et réfléchi) et sont donnés dans les équations 3.9 et 3.10 (Jordan, 1950).

$$C_h = \frac{\cos \psi - \sqrt{(\epsilon_r - jx) - \sin^2 \psi}}{\cos \psi + \sqrt{(\epsilon_r - jx) - \sin^2 \psi}} \quad (3.9)$$

$$C_v = \frac{(\epsilon_r - jx) \cos \psi - \sqrt{(\epsilon_r - jx) - \sin^2 \psi}}{(\epsilon_r - jx) \cos \psi + \sqrt{(\epsilon_r - jx) - \sin^2 \psi}} \quad (3.10)$$

$$x = \frac{\sigma}{\omega \epsilon_0} \quad (3.11)$$

ϵ_r est la permittivité relative (ou la constante diélectrique) de la surface réfléchissante, $\epsilon_0 = 8.854 \text{e-}12 \text{ F/m}$ est la permittivité du vide, σ est sa conductivité électrique, $\omega = 2\pi f$ est la pulsation de l'onde et ψ est l'angle d'incidence. On a utilisé les valeurs typiques du sol dans un premier temps ($\epsilon_r = 15$ et $\sigma = 12 \text{e-}3$). La Figure 3.2 représente les coefficients de réflexion du sol (terre arable) à une fréquence de 77 GHz en fonction de l'angle d'élévation terrestre $\psi' = 90^\circ - \psi$.

On voudrait aussi savoir comment évoluent les coefficients de réflexion quand les paramètres ϵ_r et σ varient, i.e. lorsque la surface réfléchissante n'est plus la terre. Dans notre application de poursuite automobile, connaître quelles sont les valeurs de C_h et C_v pour l'asphalte est utile puisque c'est le principal matériau des routes en général.

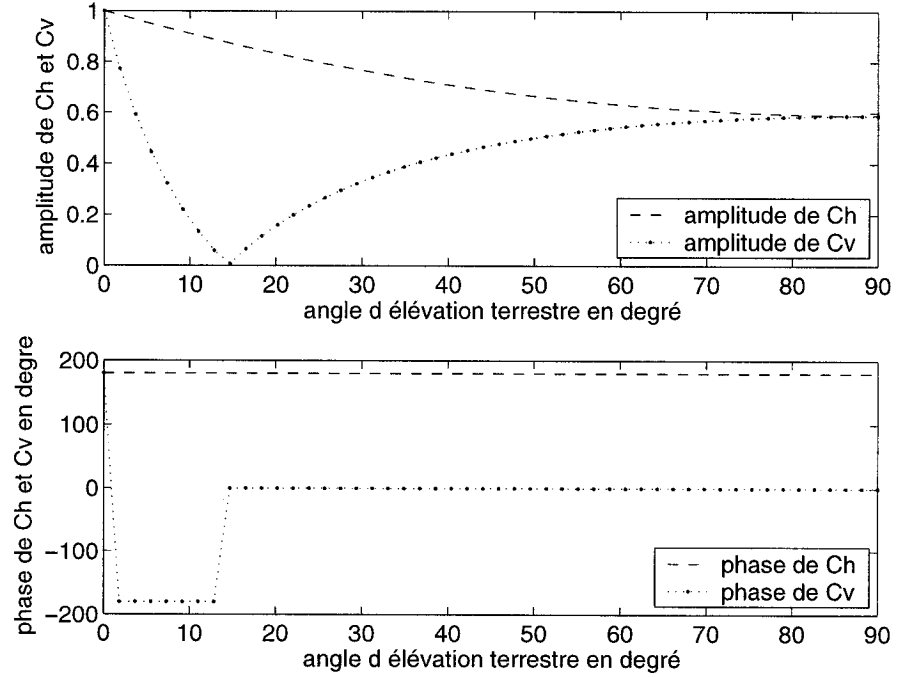


FIG. 3.2 *Coefficients de réflexion du sol en fonction de $\psi' = 90^\circ - \psi$ ($\epsilon_r = 15$, $\sigma = 12 \text{ e-}3$, $f = 77 \text{ GHz}$)*

La conductivité de l'asphalte varie avec son caractère sec ou mouillé (chaussée humide ou non). Elle est donc difficile à fixer. Une petite analyse qualitative nous montre que le poids de ce paramètre est beaucoup moins fort que celui de ϵ_r . En effet, à une fréquence de 77 GHz et pour le sol, $x = 2.80 \text{ e-}3$. Donc $x \ll \epsilon_r$ pour le sol. En identifiant la conductivité de l'asphalte à celle du sol et en prenant $\epsilon_r = 3.2$ cette fois (valeur de permittivité relative adoptée pour l'asphalte en s'inspirant du lien (Tableau de constantes diélectriques) donné dans les références), on peut simplifier les équations 3.9 et 3.10 et donner une bonne approximation des coefficients de réflexion de l'asphalte pour notre fréquence d'utilisation :

$$C_h = \frac{\cos \psi - \sqrt{\epsilon_r - \sin^2 \psi}}{\cos \psi + \sqrt{\epsilon_r - \sin^2 \psi}} \quad (3.12)$$

$$C_v = \frac{\epsilon_r \cos \psi - \sqrt{\epsilon_r - \sin^2 \psi}}{\epsilon_r \cos \psi + \sqrt{\epsilon_r - \sin^2 \psi}} \quad (3.13)$$

L'eau a une permittivité relative très élevée de l'ordre de 80. L'asphalte mouillée aura donc des propriétés réfléchissantes très différentes de l'asphalte sèche. La Figure 3.3 montre l'amplitude des coefficients de réflexion C_h et C_v en fonction de ψ' et de ϵ_r .

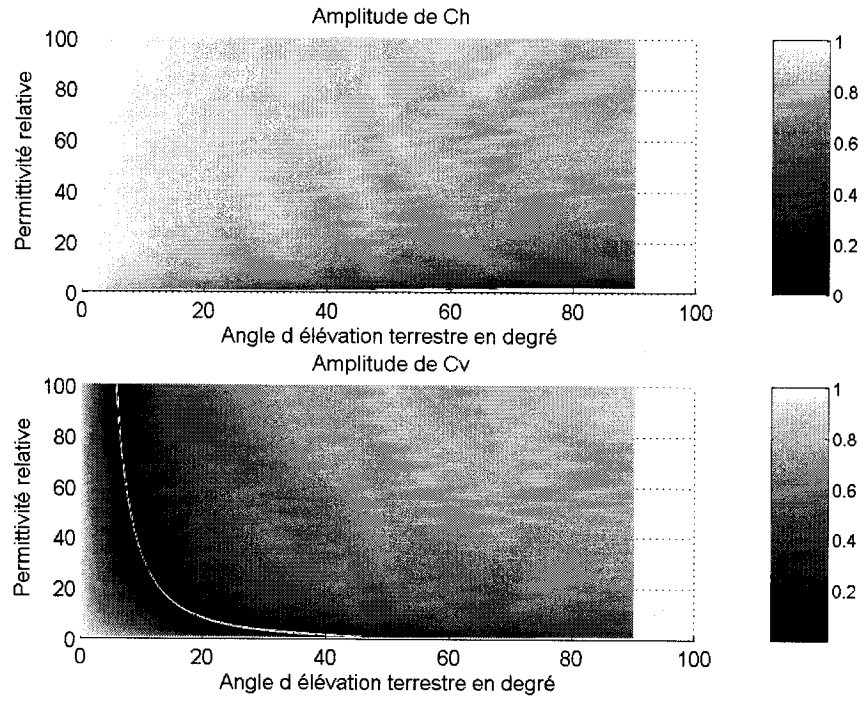


FIG. 3.3 $|C_h|$ et $|C_v|$ en fonction de ψ' et ϵ_r

On remarque que le coefficient C_v s'annule pour un certain angle d'incidence (angle de Brewster ψ_b) dépendant de la valeur de ϵ_r , i.e. du matériau et que son amplitude est toujours inférieure à celle de C_h . Cette propriété intéressante sera exploitée dans la suite.

3.2.3 Détermination expérimentale de coefficients de réflexion pour différents matériaux

Dans le dernier chapitre consacré aux résultats expérimentaux, on a utilisé des plans réflecteurs composés de différents matériaux pour générer des réflexions parasites. Pour déterminer leurs réflectivités respectives, on a créé artisanalement un banc de mesures composé de deux cornets fonctionnant à 77 GHz, l'un en émetteur et l'autre en récepteur. On a fait des tests pour des angles d'élévation de 10° , 24° , 35° et 45° . La première étape consiste à étalonner le système avec un plan métallique supposé être un réflecteur parfait. La puissance de réception relevée aux différents angles considérés nous donne la puissance reçue de référence $P_0(\psi')$. On remplace ensuite le plan métallique par un morceau de matériau i dont on veut tester la réflectivité. On mesure de même la nouvelle valeur de puissance $P_i(\psi')$ au niveau du cornet récepteur (N.B. : la directivité des cornets utilisés est importante si bien que le cornet récepteur arrive à sélectionner le faisceau réfléchi du faisceau directement émis par le premier cornet). Le rapport $\sqrt{\frac{P_i(\psi')}{P_0(\psi'')}}$ nous donne alors l'amplitude du coefficient de réflexion du matériau i . En sélectionnant le mode d'émission du cornet émetteur (plan E ou plan H), il est possible de déterminer C_h ou C_v .

La Figure 3.4 compare les valeurs théoriques avec les données empiriques obtenues selon cette méthode avec un morceau d'asphalte. On trouve une bonne correspondance entre les deux courbes. En particulier, le minimum de la courbe expérimentale se trouve entre 24° et 35° , conformément à la formule théorique qui trouve un minimum de réflexion pour $\psi' = 29.1^\circ$.

D'autres matériaux ont été testés pour valider la méthode expérimentale. Ainsi, des plaques d'asphalte, de bitume caoutchouteux, de plastique (PVC) et de bois (contre-plaqué) ont été placées sur le banc de mesure. La Figure 3.5 montre leurs coefficients de réflexion pour les quatre mêmes valeurs angulaires.

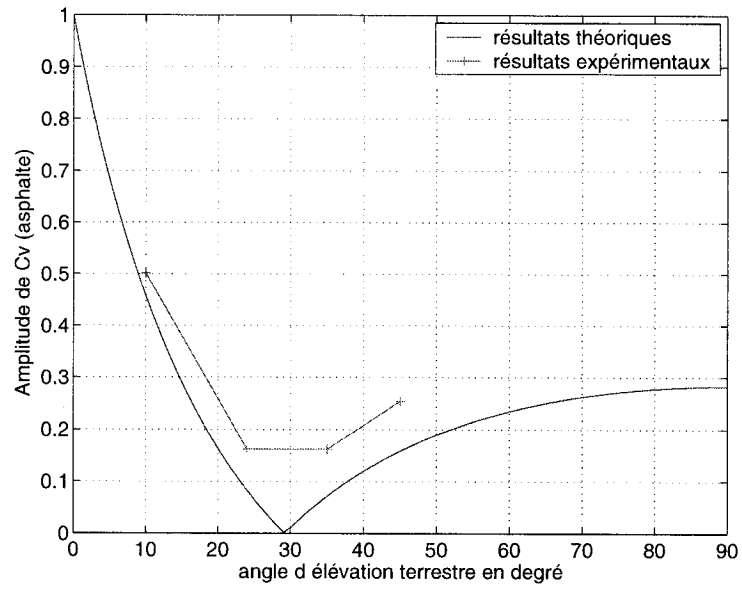


FIG. 3.4 $|C_v|$ théorique et expérimentale pour l'asphalte en fonction de ψ'

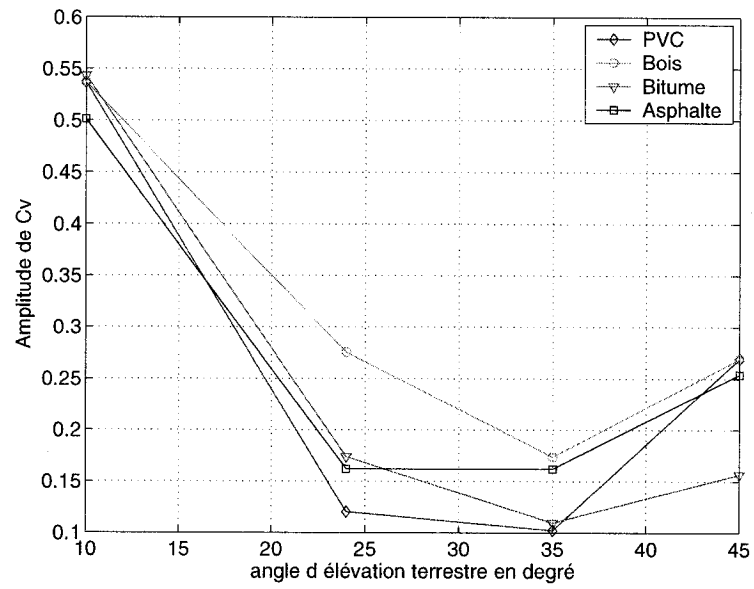


FIG. 3.5 $|C_v|$ expérimentale pour divers matériaux en fonction de ψ'

L'angle d'incidence pour lequel l'amplitude de C_v est minimale varie d'un matériau à l'autre comme on peut le vérifier ici. Cette détermination expérimentale est pratique pour avoir un ordre d'idée de la réflectivité d'un matériau composite (non homogène comme le contre-plaqué) qui est difficile à évaluer théoriquement.

Maintenant que la réflexion parasite est simulée, on passe à la modélisation des signaux perturbés par la réflexion au niveau des antennes et des combineurs de puissance.

3.2.4 Génération des signaux du système en présence de la réflexion parasite

Comme point de départ, on suppose que l'onde transmise par l'émetteur est polarisée circulairement (à droite). On suppose également que l'antenne émettrice est isotrope : son gain est unitaire pour toutes les directions de l'espace. On peut mettre le champ \vec{E} au point R sous la forme :

$$\vec{E} = \alpha \frac{e^{-2\pi jd/\lambda}}{d} (\hat{u}_v - j \hat{u}_h) \quad (3.14)$$

On définit le vecteur de polarisation unitaire \hat{p} par :

$$\hat{p} = \frac{1}{\sqrt{2}} (\hat{u}_v - j \hat{u}_h) \quad (3.15)$$

D'où finalement la nouvelle formule pour le champ \vec{E} au point R :

$$\vec{E} = \alpha' \frac{e^{-2\pi jd/\lambda}}{d} \hat{p} \quad (3.16)$$

avec $\alpha' = \sqrt{2}\alpha$.

Pour que la réception soit adaptée entre les deux antennes, la polarisation des antennes du système de réception doit être également circulaire droite. On peut reprendre les équations 2.3 du chapitre précédent en tenant compte de la polarisation cette fois. Le gain des antennes de réception est encore en $\cos \theta$. On appelle toujours A_{x1} , A_{x2} et A_{x3} les signaux de sortie des antennes 1,2 et 3 respectivement du réseau d'axe X . Les équations 3.17 en donnent les formules.

$$A_{x1} = G(\theta)\alpha'(\hat{p} \cdot \hat{p}^*) \frac{e^{-2\pi j d/\lambda}}{d} \quad (3.17a)$$

$$A_{x2} = G(\theta)\alpha'(\hat{p} \cdot \hat{p}^*) \frac{e^{-2\pi j d/\lambda}}{d} \exp(2\pi j \cdot 0.58 \cdot \cos \theta_x) \quad (3.17b)$$

$$A_{x3} = G(\theta)\alpha'(\hat{p} \cdot \hat{p}^*) \frac{e^{-2\pi j d/\lambda}}{d} \exp(2\pi j \cdot 1.37 \cdot \cos \theta_x) \quad (3.17c)$$

Comme les antennes en R sont à polarisation circulaire droite, leur vecteur de polarisation vaut \hat{p} dans un repère $(\hat{u}'_r, \hat{u}'_h, \hat{u}'_v)$ avec $\hat{u}'_r = -\hat{u}_r$, $\hat{u}'_h = \hat{u}_h$ et $\hat{u}'_v = -\hat{u}_v$. Leur vecteur de polarisation vaut donc en fait \hat{p}^* dans le même repère $(\hat{u}_r, \hat{u}_h, \hat{u}_v)$ que l'émetteur E . La transmission entre les deux antennes est alors adaptée : le facteur d'adaptation $\hat{p} \cdot \hat{p}^*$ vaut 1. On retrouve donc exactement les équations 2.3 à une constante multiplicative près.

On rajoute maintenant le signal réfléchi dans le modèle. On appelle $A_{x1,r}$, $A_{x2,r}$ et $A_{x3,r}$ les signaux de sortie perturbés par la réflexion des antennes 1,2 et 3 respectivement du réseau d'axe X . Les équations 3.19 en donnent les formules. L'indice r représente le signal réfléchi.

Le facteur $\frac{C_v(\psi) + C_h(\psi)}{2}$ qui pondère l'amplitude et la phase du signal réfléchi provient de l'adaptation entre le faisceau réfléchi polarisé elliptiquement cette fois et les antennes réceptrices à polarisation circulaire. Il provient de l'équation 3.18.

$$\frac{C_v \hat{u}_v - j C_h \hat{u}_h}{\sqrt{2}} \cdot \hat{p}^* = \frac{C_v + C_h}{2} \quad (3.18)$$

$$A_{x1,r} = G(\theta)\alpha' \frac{e^{-2\pi jd/\lambda}}{d} + G(\theta_r) \frac{C_v(\psi) + C_h(\psi)}{2} \alpha' \frac{e^{-2\pi jd_r/\lambda}}{d_r} \quad (3.19a)$$

$$A_{x2,r} = G(\theta)\alpha' \frac{e^{-2\pi jd/\lambda}}{d} \exp(2\pi j \cdot 0.58 \cdot \cos \theta_x) \\ + G(\theta_r) \frac{C_v(\psi) + C_h(\psi)}{2} \alpha' \frac{e^{-2\pi jd_r/\lambda}}{d_r} \exp(2\pi j \cdot 0.58 \cdot \cos \theta_{x,r}) \quad (3.19b)$$

$$A_{x3,r} = G(\theta)\alpha' \frac{e^{-2\pi jd/\lambda}}{d} \exp(2\pi j \cdot 1.37 \cdot \cos \theta_x) \\ + G(\theta_r) \frac{C_v(\psi) + C_h(\psi)}{2} \alpha' \frac{e^{-2\pi jd_r/\lambda}}{d_r} \exp(2\pi j \cdot 1.37 \cdot \cos \theta_{x,r}) \quad (3.19c)$$

On peut également écrire les équations 3.20 donnant les signaux de sortie $A_{y1,r}$, $A_{y2,r}$ et $A_{y3,r}$ perturbés par la réflexion des antennes 1,2 et 3 respectivement du réseau d'axe Y .

$$A_{y1,r} = G(\theta)\alpha' \frac{e^{-2\pi jd/\lambda}}{d} + G(\theta_r) \frac{C_v(\psi) + C_h(\psi)}{2} \alpha' \frac{e^{-2\pi jd_r/\lambda}}{d_r} \quad (3.20a)$$

$$A_{y2,r} = G(\theta)\alpha' \frac{e^{-2\pi jd/\lambda}}{d} \exp(2\pi j \cdot 0.58 \cdot \cos \theta_y) \\ + G(\theta_r) \frac{C_v(\psi) + C_h(\psi)}{2} \alpha' \frac{e^{-2\pi jd_r/\lambda}}{d_r} \exp(2\pi j \cdot 0.58 \cdot \cos \theta_{y,r}) \quad (3.20b)$$

$$A_{y3,r} = G(\theta)\alpha' \frac{e^{-2\pi jd/\lambda}}{d} \exp(2\pi j \cdot 1.37 \cdot \cos \theta_y) \\ + G(\theta_r) \frac{C_v(\psi) + C_h(\psi)}{2} \alpha' \frac{e^{-2\pi jd_r/\lambda}}{d_r} \exp(2\pi j \cdot 1.37 \cdot \cos \theta_{y,r}) \quad (3.20c)$$

Les angles $\theta_{x,r}$ et $\theta_{y,r}$ sont donnés par les équations 3.21.

$$\cos \theta_{x,r} = \sin \theta_r \cos \phi_r \quad (3.21a)$$

$$\cos \theta_{y,r} = \sin \theta_r \sin \phi_r \quad (3.21b)$$

On génère ensuite des signaux de puissance en sortie des combineurs, perturbés par la réflexion et entachés de bruit. On reprend donc les équations 2.26 pour MLPx

qui deviennent les équations 3.22, tenant compte de la réflexion parasite.

$$x_{1x,r} = -j \frac{A_{x1,r} - A_{x2,r}}{2} \quad (3.22a)$$

$$x_{2x,r} = -\frac{A_{x1,r} + A_{x2,r}}{2} \quad (3.22b)$$

$$x_{3x,r} = -\frac{jA_{x2,r} + A_{x3,r}}{2} \quad (3.22c)$$

$$x_{4x,r} = -\frac{A_{x2,r} + jA_{x3,r}}{2} \quad (3.22d)$$

$$x_{5x,r} = -j \frac{A_{x1,r} - A_{x3,r}}{\sqrt{8}} \quad (3.22e)$$

$$x_{6x,r} = -\frac{A_{x1,r} + A_{x3,r}}{\sqrt{8}} \quad (3.22f)$$

$$x_{7x,r} = -\frac{jA_{x1,r} + A_{x3,r}}{\sqrt{8}} \quad (3.22g)$$

$$x_{8x,r} = -\frac{A_{x1,r} + jA_{x3,r}}{\sqrt{8}} \quad (3.22h)$$

On rajoute ensuite le bruit complexe d'amplitude gaussienne sur ces signaux avant d'en prendre l'amplitude au carré pour obtenir les signaux de puissance normalisés et numérisés formant les vecteurs d'entrée de MLPx (équations similaires pour MLPy). On reprend les mêmes formules que dans le chapitre précédent pour aboutir aux équations 3.23.

$$X_{ix,r}^b = \|x_{ix,r} + n_{ix}\|^2, \quad i = 1, \dots, 8 \quad (3.23a)$$

$$X_{ix,r,n}^b = \frac{X_{ix,r}^b}{\max_{j=1,\dots,8} X_{jx,r}^b}, \quad i = 1, \dots, 8 \quad (3.23b)$$

3.3 Simulation de la robustesse du système à une réflexion parasite et au bruit

Ces signaux vont nous aider à tester la robustesse de notre système (MLPx et MLPy dans un premier temps) à une réflexion parasite générée par une route plane en

asphalte (dont le ϵ_r est estimé à 3.2). Pour cela, on utilise les deux réseaux MLPx et MLPy déjà entraînés par la méthode décrite à la fin du chapitre précédent. Les points d'entraînement sont ceux d'un échantillonnage équiréparti sur une calotte sphérique définie par le cône d'intérêt. Pour les points de validation, les DOAs sont celles situées au centre des carreaux du maillage des points d'entraînement. On va générer tous les signaux $X_{ix,r,n}^b$ et $X_{iy,r,n}^b$, $i = 1, \dots, 8$ correspondant à ces DOAs du signal principal et on va comparer les sorties rendues par MLPx et MLPy aux vrais angles θ_x et θ_y . Le RSB est toujours de 20 dB, la fréquence de travail de 5.8 GHz ($\lambda = 5.17 \text{ cm}$).

Par contre, on va prendre une distance $d = 10 \text{ m}$ pour la phase de validation. En effet, en pratique, cette distance sera largement supérieure à $d = \sqrt{2}m$ (valeur prise pour l'entraînement). Cette valeur particulière a été adoptée parce qu'elle permet de générer toutes les DOAs du cône d'intérêt d'ouverture 90° . En prenant une distance d plus grande, certaines DOAs vont correspondre à des émetteurs placés sous terre, ce qui n'est pas possible. Le cône d'intérêt est donc tronqué. Seuls les points vérifiant $\overrightarrow{RE} \cdot \hat{y} > -h$ vont correspondre à des DOAs réelles (émetteur situé dans l'air ambiant). On ne prendra pas une distance d trop grande non plus de manière à générer sensiblement plus que la moitié du nombre de DOAs total du cône d'intérêt. Avec $d = 10 \text{ m}$ et $h = 1 \text{ m}$, on génère 58 % du nombre de DOAs exactement. Les Figures 3.6, 3.7 et 3.8 exposent les résultats de simulation avec les critères précédents.

Les effets néfastes du bruit et de la réflexion parasite sont superposés sur les figures. Pour isoler ceux de la réflexion seulement, on va appliquer un pré-traitement sur 30 échantillons de façon à réduire drastiquement ceux du bruit. Les Figures 3.9, 3.10 et 3.11 montrent principalement les effets de la réflexion sur la qualité de l'estimation.

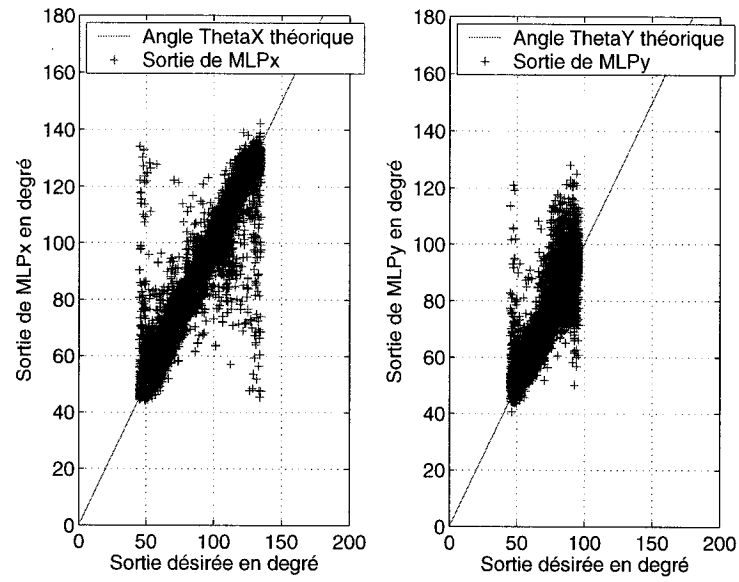


FIG. 3.6 Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite, avec une réflexion parasite

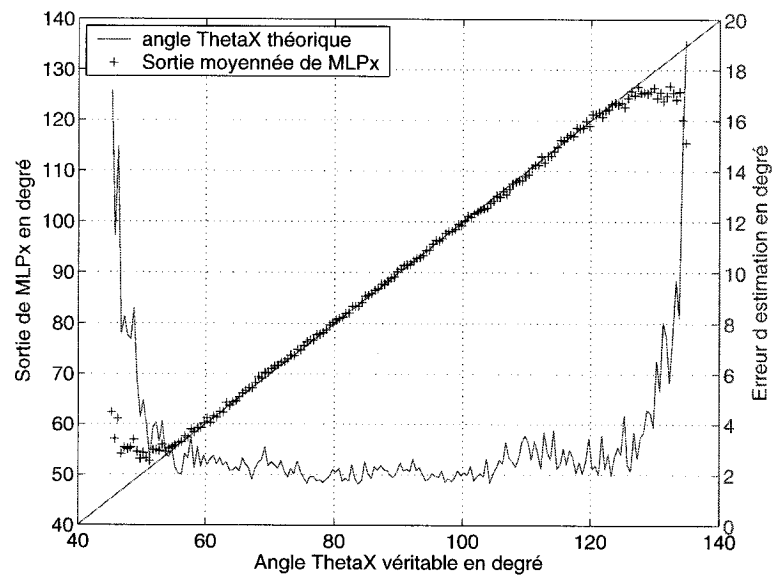


FIG. 3.7 Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx avec une réflexion parasite

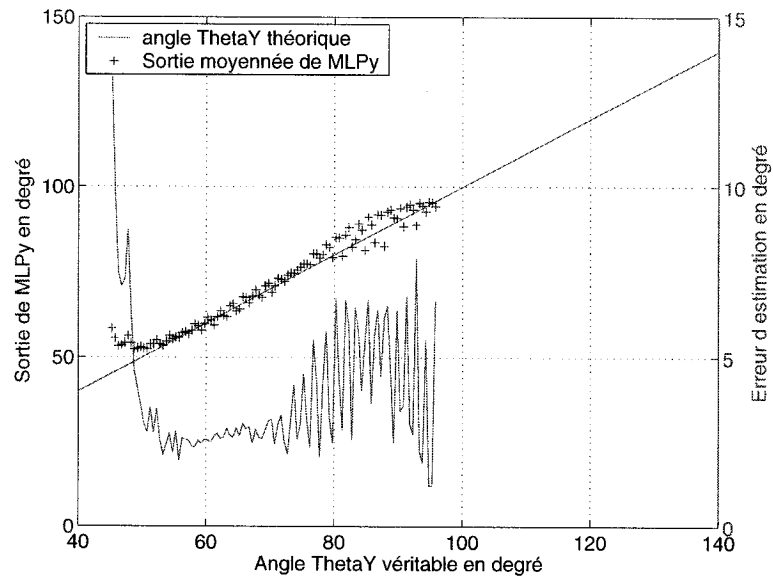


FIG. 3.8 Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec une réflexion parasite

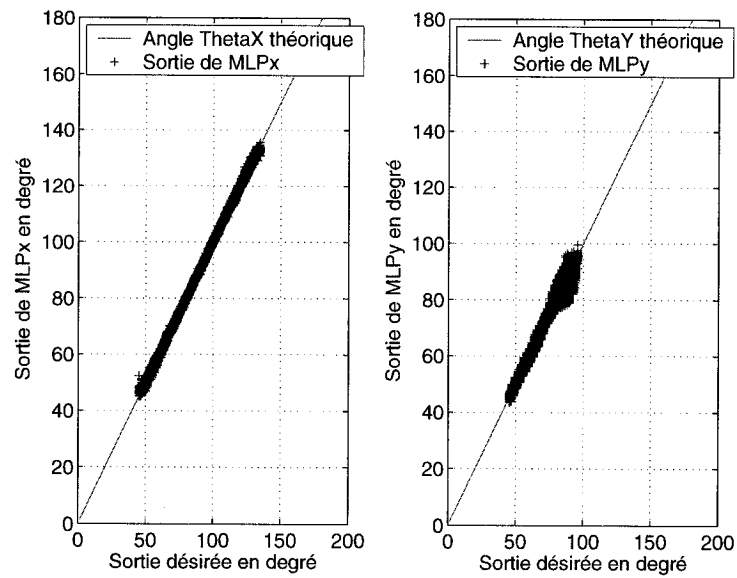


FIG. 3.9 Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite, avec une réflexion parasite et 30 échantillons de pré-traitement

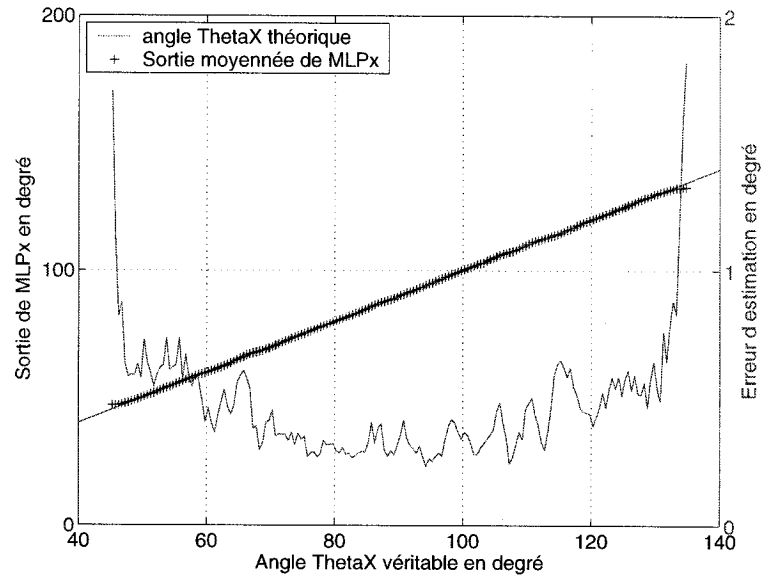


FIG. 3.10 *Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx avec une réflexion parasite et 30 échantillons de pré-traitement*

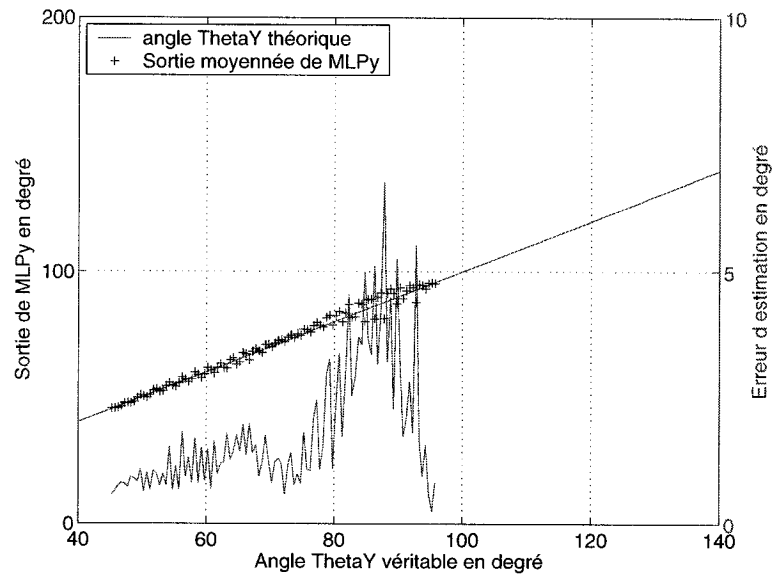


FIG. 3.11 *Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec une réflexion parasite et 30 échantillons de pré-traitement*

Le Tableau 3.1 récapitule les résultats obtenus selon les différentes définitions d'erreurs (les mêmes que dans le chapitre précédent), sous un niveau de bruit de -20 dB et avec une réflexion parasite générée par une route plane en asphalte. La troisième ligne du tableau est celle obtenue sans pré-traitement, sous un bruit de RSB fixé à 35 dB.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}
RSB = 20 dB et $N_{pre} = 1$	2.71°	3.77°	5.35°	8.21°	5.32°
RSB = 20 dB et $N_{pre} = 30$	0.42°	2.02°	2.14°	2.74°	2.35°
RSB = 35 dB et $N_{pre} = 1$	0.42°	2.02°	2.14°	2.75°	2.35°

TAB. 3.1 *Comparaison des résultats effectués sur des signaux bruités perturbés par une réflexion, avec et sans pré-traitement*

On remarque qu'un niveau de bruit de -20 dB avec $N_{pre} = 30$ donne des résultats identiques que sous un niveau de bruit de -35 dB sans pré-traitement. On peut expliquer facilement cette coïncidence en utilisant le théorème de la limite centrale. Soit σ_{eq}^2 la variance de l'amplitude du bruit équivalent en sortie du filtre de pré-traitement.

$$\sigma_{eq}^2 = \frac{\sigma^2}{N_{pre}} \quad (3.24)$$

σ^2 est la variance de l'amplitude du bruit complexe gaussien pour chaque échantillon en amont du filtre. Comme la variance d'un bruit gaussien est aussi sa puissance, le RSB_{eq} en sortie du filtre est donnée par :

$$RSB_{eq} = RSB + 10 \log(N_{pre}) \quad (3.25)$$

avec RSB le rapport signal à bruit de chaque échantillon en amont du filtre. Si on prend $RSB = 20dB$ et $N_{pre} = 30$, on obtient $RSB_{eq} = 34.77dB$. Ce n'est donc pas

un hasard si les deux dernières lignes du Tableau 3.1 donnent les mêmes résultats. Dans la suite, on prendra souvent un *RSB* de 35 dB directement. Les simulations sont beaucoup plus rapides quand on n'a pas de pré-traitement à effectuer. De plus, avec un *RSB* si bon, le bruit devient négligeable devant l'effet de la réflexion parasite. On peut donc mieux étudier la robustesse du système à la réflexion de cette façon.

Le tableau précédent nous montre que θ_x est beaucoup moins perturbé que θ_y par la réflexion. Ceci s'explique par le fait que $|\theta_x - \theta_{x,r}|$ est très faible pour la plupart des DOAs du cône d'intérêt, contrairement à $|\theta_y - \theta_{y,r}|$. Pour le réseau d'antennes d'axe X , tout se passe comme s'il ne recevait qu'un seul signal, plus ou moins puissant que le faisceau principal, mais ayant le même θ_x . Ce n'est pas le cas pour le réseau d'antennes d'axe Y . En regardant attentivement la Figure 3.11, on remarque que la courbe décrivant l'erreur angulaire d'estimation forme deux bosses, le creux entre les deux correspondant à des DOAs pour lesquelles les coefficients de Fresnel (C_v principalement) ont une amplitude faible.

Dans les prochaines sections, on va tester différentes structures pour essayer de diminuer l'erreur angulaire d'estimation en θ_y surtout.

3.4 Intérêt du découplage de θ_x et θ_y

Dans (Coudyser, 2003), deux architectures neuronales différentes étaient comparées : celle avec deux MLPs (MLPx et MLPy) décrits ici depuis le début, ou celle faisant appel à un seul MLP, ayant une couche cachée également, prenant des vecteurs de 16 composantes en entrée (les mêmes que MLPd) et rendant un vecteur de deux composantes en sortie, soit les angles θ_x et θ_y .

On entraîne donc un MLP composé de 30 neurones sur sa couche cachée, pendant

30 époques. Par rapport aux travaux d'optimisation précédents, cette structure, même si elle n'est pas optimisée selon notre méthode habituelle (optimisation du nombre de neurones dans la couche cachée à nombre d'époques d'entraînement fixé, 30 typiquement, puis optimisation du nombre d'époques avec le nombre de neurones trouvé), va donner de très bons résultats. En effet, l'erreur d'entraînement a déjà convergé au bout de 30 époques quand on utilise la fonction "trainlm" de la Neural Networks Toolbox de Matlab[®] avec la matrice de points d'entraînement considérée. On cherchera à l'optimiser dans la suite si d'aventure on s'aperçoit qu'il est plus judicieux d'avoir recours à un seul gros MLP plutôt que MLPx et MLPy. Les antennes sont toujours à polarisation circulaire. Pour la phase d'entraînement, le bruit est de -20 dB, $h = 1\text{ m}$ et $d = \sqrt{2}\text{ m}$. Pour la phase de validation, le bruit est de -35 dB (et $N_{pre} = 1$), $h = 1\text{ m}$, $d = 10\text{ m}$ et une réflexion parasite est générée par une route plane d'asphalte ($\epsilon_r = 3.2$). La fréquence de travail est de 5.8 GHz. Le Tableau 3.2 montre les résultats ainsi obtenus.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}
RSB = 35 dB et $N_{pre} = 1$	0.83°	2.02°	2.37°	2.91°	2.56°

TAB. 3.2 Résultats effectués sur des signaux bruités perturbés par une réflexion en utilisant une architecture à un seul MLP

On remarque que l'erreur d'estimation en θ_x est deux fois plus élevée que celle calculée par MLPx dans les mêmes conditions. Celle pour θ_y est la même. Cette dégradation des performances en θ_x se comprend aisément : la perturbation de la réflexion est présente surtout sur les 8 composantes $X_{iy,r,n}^b$, $i = 1, \dots, 8$ du vecteur d'entrée. Elle se répercute nécessairement sur le vecteur de sortie, i.e. sur θ_y mais aussi, chose nouvelle et non désirée, sur θ_x . La Figure 3.12 représente les différences angulaires.

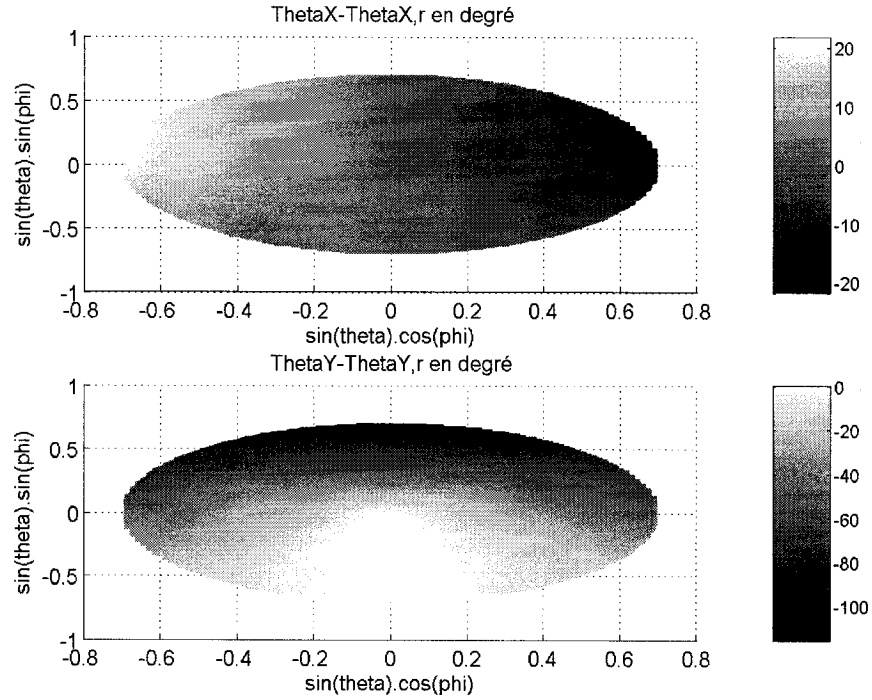


FIG. 3.12 Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite, avec une réflexion parasite et 30 échantillons de pré-traitement

Conclusion : contrairement à (Coudyser, 2003) où une structure à un seul MLP était finalement adoptée pour faire face au bruit, on retient ici une structure à deux RNAs, MLPx et MLPy, pour estimer séparément θ_x et θ_y en présence de bruit et d'une réflexion parasite.

3.5 Etude de la polarisation des antennes

On s'intéresse maintenant aux éléments du réseau d'antennes. Des antennes à polarisation circulaire avaient été retenues dans (Coudyser, 2003) et (Fonseca, 2003).

Des antennes à spirale d'Archimède et à fente en croix inégale avaient été testées. De telles antennes sont difficiles à mettre en œuvre. Pour cette raison, on va tester des antennes à polarisation linéaire, beaucoup plus simple à fabriquer. Ainsi, nos résultats expérimentaux du dernier chapitre ont été obtenus avec de simples guides d'onde tronqués. On a le choix entre des antennes à polarisation horizontale ou verticale.

On donne donc les équations 3.26 régissant les signaux de sortie des antennes à polarisation horizontale en présence d'une réflexion parasite.

$$A_{x1,r}^h = G(\theta)\alpha \frac{e^{-2\pi jd/\lambda}}{d} + G(\theta_r)C_h \alpha \frac{e^{-2\pi jd_r/\lambda}}{d_r} \quad (3.26a)$$

$$A_{x2,r}^h = G(\theta)\alpha \frac{e^{-2\pi jd/\lambda}}{d} \exp(2\pi j \cdot 0.58 \cdot \cos \theta_x) + G(\theta_r)C_h \alpha \frac{e^{-2\pi jd_r/\lambda}}{d_r} \exp(2\pi j \cdot 0.58 \cdot \cos \theta_{x,r}) \quad (3.26b)$$

$$A_{x3,r}^h = G(\theta)\alpha \frac{e^{-2\pi jd/\lambda}}{d} \exp(2\pi j \cdot 1.37 \cdot \cos \theta_x) + G(\theta_r)C_h \alpha \frac{e^{-2\pi jd_r/\lambda}}{d_r} \exp(2\pi j \cdot 1.37 \cdot \cos \theta_{x,r}) \quad (3.26c)$$

Les équations 3.27 établissent les signaux de sortie des antennes à polarisation verticale en présence d'une réflexion parasite.

$$A_{x1,r}^v = G(\theta)\alpha \frac{e^{-2\pi jd/\lambda}}{d} + G(\theta_r)C_v \alpha \frac{e^{-2\pi jd_r/\lambda}}{d_r} \quad (3.27a)$$

$$A_{x2,r}^v = G(\theta)\alpha \frac{e^{-2\pi jd/\lambda}}{d} \exp(2\pi j \cdot 0.58 \cdot \cos \theta_x) + G(\theta_r)C_v \alpha \frac{e^{-2\pi jd_r/\lambda}}{d_r} \exp(2\pi j \cdot 0.58 \cdot \cos \theta_{x,r}) \quad (3.27b)$$

$$A_{x3,r}^v = G(\theta)\alpha \frac{e^{-2\pi jd/\lambda}}{d} \exp(2\pi j \cdot 1.37 \cdot \cos \theta_x) + G(\theta_r)C_v \alpha \frac{e^{-2\pi jd_r/\lambda}}{d_r} \exp(2\pi j \cdot 1.37 \cdot \cos \theta_{x,r}) \quad (3.27c)$$

Ces signaux sont ensuite combinés, normalisés et numérisés de la même façon que les autres à polarisation circulaire. On réentraîne de nouveau MLPx et MLPy avec 30 neurones chacun dans leur couche cachée, pendant 30 époques, sous un RSB de 20 dB, avec $h = 1\text{ m}$ et $d = \sqrt{2}\text{ m}$, sans réflexion. On teste ensuite la structure entraînée (aussi bien les antennes à polarisation horizontale que verticale) avec des signaux perturbés par une réflexion générée par une route en asphalte, sous un RSB de 35 dB (sans pré-traitement donc), avec $h = 1\text{ m}$ et $d = 10\text{ m}$. Le Tableau 3.3 montre les résultats que l'on obtient.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}
polarisation horizontale	0.56°	3.79°	3.93°	5.08°	4.21°
polarisation verticale	0.39°	1.17°	1.32°	1.71°	1.41°

TAB. 3.3 *Résultats effectués sur des signaux bruités perturbés par une réflexion en utilisant des antennes à polarisation horizontale ou verticale, deux RNAs MLPx et MLPy*

Conclusion : utiliser des antennes à polarisation verticale (cas 3) réduit de manière considérable l'erreur d'estimation en θ_y par rapport aux antennes à polarisation circulaire (cas 1), ou pire, à polarisation horizontale (cas 2). Ceci s'explique par l'amplitude de la réflexion, pondérée par le facteur $\frac{C_h + C_v}{2}$ (cas 1), C_h (cas 2) ou C_v (cas 3). Or, $|C_v(\psi)| < \left| \frac{C_h(\psi) + C_v(\psi)}{2} \right| < |C_h(\psi)| \quad \forall \psi < \psi_b$. Pour $\epsilon_r = 3.2$, cette inégalité reste vraie tant que $\psi < 42^\circ$, avec $\psi_b = 29^\circ$ (voir Figure 3.13). Or, à la limite du cône d'intérêt, $\psi = 52^\circ$ (dans le plan (Y,Z) et avec $d = 10\text{ m}$). Pour la majorité des situations, on a $\psi < 42^\circ$ malgré tout, ce qui explique qu'une polarisation verticale donne de meilleurs résultats qu'une polarisation circulaire. Surtout, ψ est faible pour des DOAs situées autour de l'axe Z , i.e. les DOAs pour lesquelles une très bonne estimation est recherchée.

Dans toute la suite du mémoire, on utilisera donc des antennes à polarisation verticale.

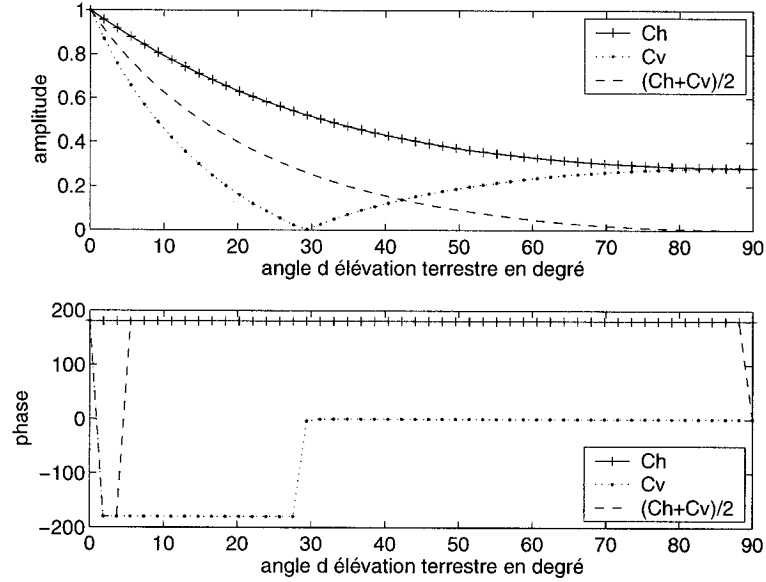


FIG. 3.13 *Coefficients de réflexion intervenant dans des polarisations horizontale, verticale et circulaire*

3.6 Comparaisons avec un réseau plus simple à deux paires d'antennes

Dans (Coudyser, 2003), deux structures différentes pour le système étaient comparées, du point de vue robustesse au bruit. La structure basée sur deux réseaux de trois antennes chacun et une autre formée par deux réseaux de deux antennes, plus simple. Finalement, c'est ce dernier modèle qui avait été adopté, parce qu'il donnait de meilleurs résultats en terme d'erreur angulaire sur la DOA en présence de bruit.

On ne va pas redonner les équations des signaux de sortie des antennes et des combineurs de signaux, ils sont donnés dans (Coudyser, 2003). Le principe d'ob-

tention est le même que celui décrit dans le chapitre précédent de ce mémoire, on ne s'y attardera donc pas. Les deux signaux de sortie des antennes d'axe X (respectivement Y) vont être combinés pour donner trois signaux de puissance qui vont former le vecteur d'entrée de MLPx (MLPy respectivement).

Une optimisation rapide de MLPx et MLPy nous montre qu'il est suffisant de prendre 20 neurones dans la couche cachée pour être proche de la meilleure structure neuronale. On les entraîne chacun pendant 30 époques. Les antennes sont à polarisation verticale et ont un gain en $\cos\theta$. 30000 vecteurs environ forment les matrices d'entraînement et de validation (on utilise un quadrillage initial de 200×200 points pour générer les DOAs). Pendant l'entraînement, le RSB est fixé à 20 dB, $h = 1$ m et $d = \sqrt{2}$ m. Pour la phase de validation avec une réflexion parasite générée par une route en asphalte ($\epsilon_r = 3.2$), le RSB vaut 35 dB, $h = 1$ m et $d = 10$ m. Les Figures 3.14, 3.15 et 3.16 montrent les résultats que l'on obtient avec un système à 4 antennes entraîné et testé de la sorte.

Le Tableau 3.4 expose les différentes erreurs d'estimation faites par ce nouveau système et compare ses performances avec la paire de réseaux à trois antennes vue précédemment.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}
réseau 2×2 antennes	0.93°	1.86°	2.30°	2.74°	2.32°
gain p/r au réseau 2×3 antennes	-0.54°	-0.69°	-0.98°	-1.03°	-0.91°

TAB. 3.4 Résultats effectués sur des signaux bruités perturbés par une réflexion en utilisant un réseau à deux paires d'antennes, deux RNAs MLPx et MLPy

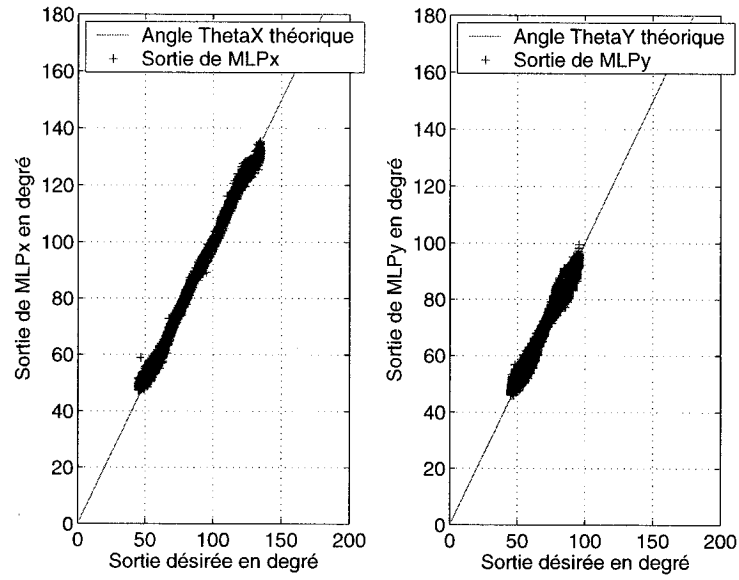


FIG. 3.14 Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite, avec un réseau à deux paires d'antennes

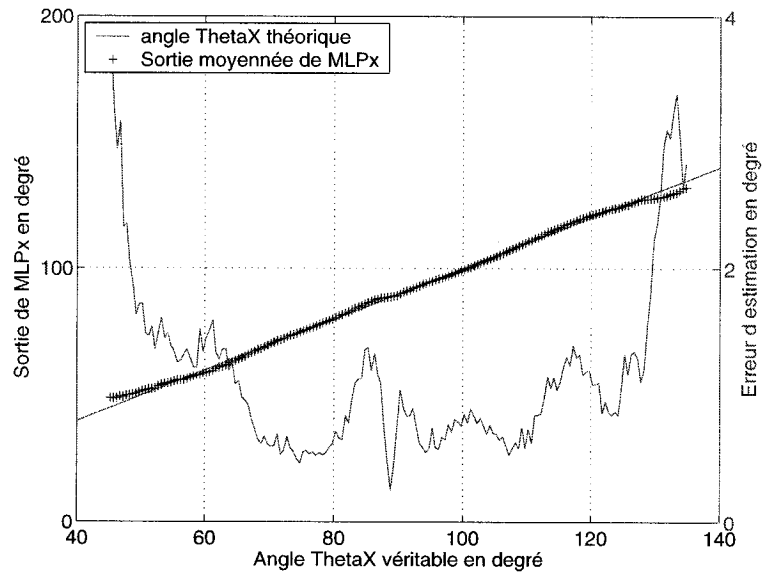


FIG. 3.15 Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx avec un réseau à deux paires d'antennes

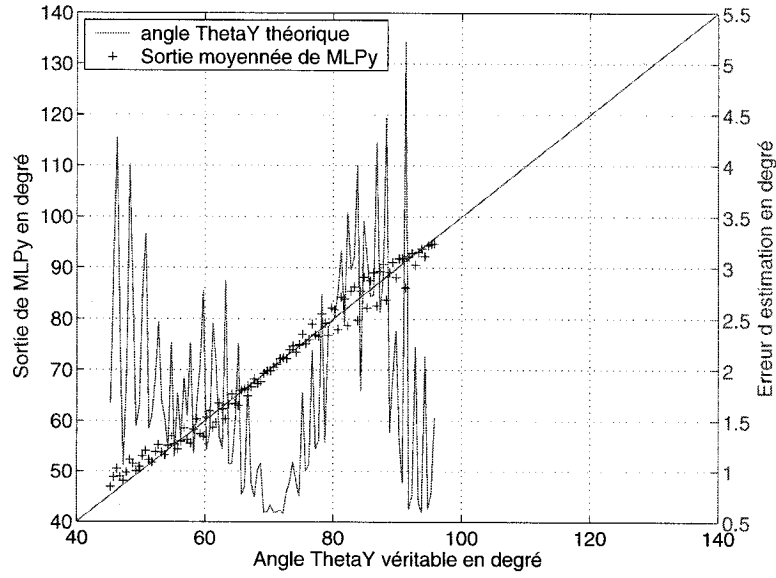


FIG. 3.16 *Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec un réseau à deux paires d'antennes*

Conclusion : utiliser un réseau plus simple donc moins onéreux se fait au détriment de la qualité d'estimation de la DOA. Contrairement à (Coudyser, 2003) qui montrait qu'un système à deux paires d'antennes était plus robuste au bruit qu'un système à deux triplets d'antennes, il apparaît ici qu'il est plus judicieux de garder notre système à six antennes, dépeint tout au long de ce mémoire, quand on veut qu'il soit aussi robuste à une réflexion parasite.

3.7 Performances de MLPd

3.7.1 Présentation de l'étage de détection

On décrit ici le fonctionnement de MLPd et son entraînement. Cet étage consiste à distinguer les signaux appartenant au cône d'intérêt de ceux situés en dehors.

Dans (Coudyser, 2003), la sortie de MLPd vaut 1 si le signal reçu appartient au cône d'intérêt d'ouverture 90° et 0 dans le cas contraire. MLPd est un MLP à une couche cachée. Ses fonctions de transfert sont des sigmoïdes pour les neurones de la couche cachée et une fonction linéaire pour le neurone de sortie. On définit un seuil à 0.5 au dessus duquel la sortie vaut 1 et en dessous duquel la sortie vaut 0. On est obligé de passer par cet intermédiaire d'une fonction linéaire pour entraîner MLPd. Si on utilisait directement une fonction échelon en sortie de MLPd, l'algorithme de rétropropagation d'erreur serait pris en défaut puisqu'il repose sur la dérivabilité des fonctions de transfert.

Il est important de rappeler ici la nécessité d'utiliser un RNA prenant la totalité des signaux de puissance pour détecter l'appartenance ou non du signal au cône d'intérêt. On pourrait penser qu'il suffit de prendre un nombre restreint de ces signaux pour effectuer ce travail pour peu qu'ils soient bien choisis. On remarque qu'il n'est pas possible de trouver un classificateur simple basé sur la donnée de seulement deux signaux de puissance normalisés parmi $X_{ix,n}$ ou $X_{iy,n}$. En effet, on a tracé les sorties des points d'entraînement servant à entraîner MLPd en fonction de deux des huit signaux de puissance pour l'axe X d'une part (Figure 3.17), et pour l'axe Y d'autre part (Figure 3.18).

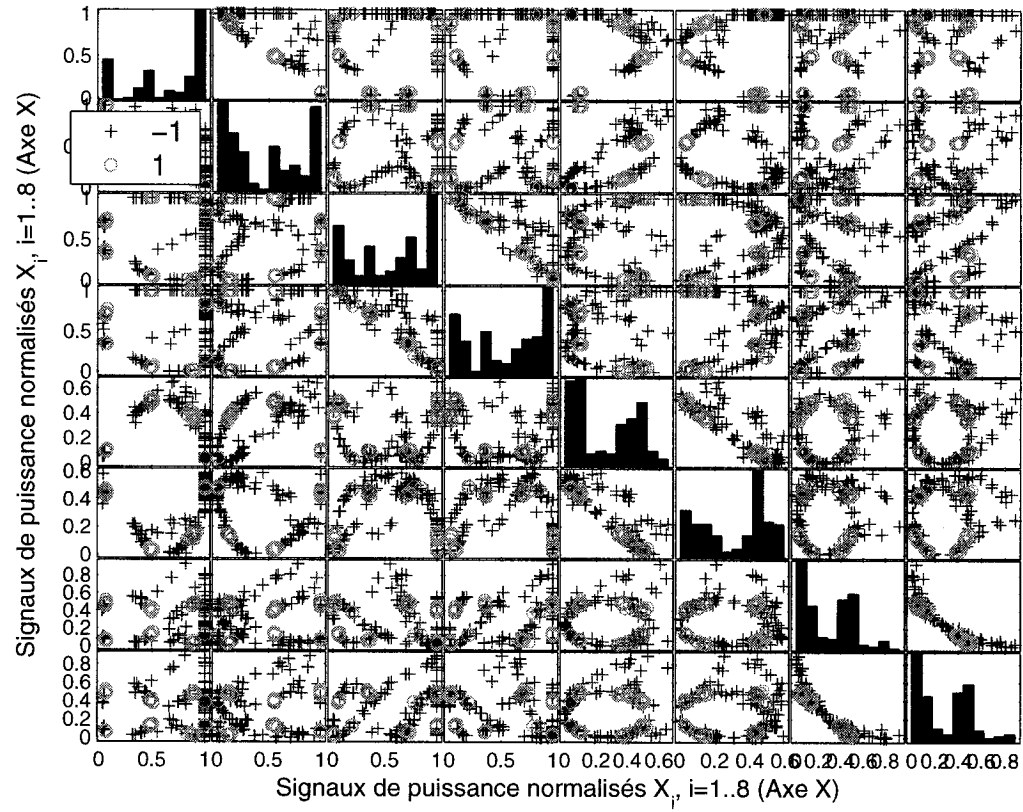


FIG. 3.17 *Etude de la complexité du classificateur en étudiant la sortie de MLPd en fonction des signaux de puissance du combineur, deux à deux (axe X)*

Cela représente 28 graphiques différents pour chaque axe. Nous n'avons pas tracé les autres graphes qui utilisent un signal d'axe X et un autre d'axe Y . On arrive à 112 ($= \frac{16^2}{2} - 16$) figures différentes en tout. Mais dans tous les cas, aucun classificateur évident n'apparaît. La donnée de deux composantes n'est donc pas suffisante pour déterminer l'appartenance du signal au cône d'intérêt. On considère dans la suite qu'en prenant les 16 composantes de puissance en entrée, on maximise nos chances de classer correctement le signal d'arrivée.

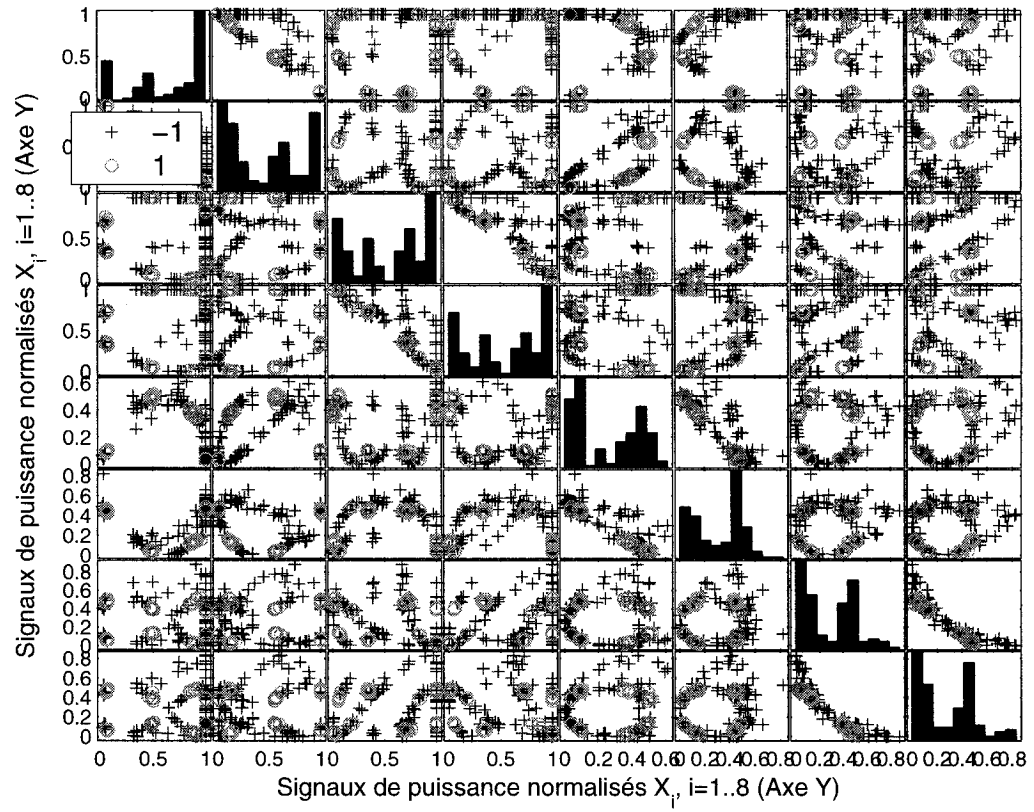


FIG. 3.18 *Etude de la complexité du classificateur en étudiant la sortie de MLPd en fonction des signaux de puissance du combineur, deux à deux (axe Y)*

Ainsi, MLPd prend un vecteur de 16 composantes en entrée, ceux des combineurs de puissance de l'axe X et de l'axe Y . On prend l'erreur quadratique moyenne comme fonction coût, soit J_{mse} . On définit un seuil de puissance pour nos simulations. En effet, comme le gain des antennes est en $\cos \theta$, aucun signal n'est reçu par le système pour des faisceaux arrivant sous un angle $\theta = 90^\circ$. On ne peut donc entraîner le RNA avec de tels signaux (apprentissage du bruit seul). C'est aussi le cas pour des angles inférieurs à 90° , mais proches de cette valeur. La règle est que si le signal reçu est de l'ordre de grandeur du bruit, il est difficile de faire un bon

apprentissage. L'astuce consiste à se fixer un seuil de puissance S_p . Si le signal reçu est supérieur à S_p , on fait le travail de détection mais par contre s'il est inférieur à S_p , on l'ignore totalement. Aucun travail n'est fait jusqu'à réception d'un signal plus fort. On a besoin de définir la puissance moyenne du signal P_m pour une DOA donnée :

$$P_m = \frac{1}{16} \sum_{i=1}^8 x_{ix}^2 + x_{iy}^2 \quad (3.28)$$

P_m est fonction de θ_x , θ_y et d , la distance émetteur-récepteur avec en particulier $P_m \propto \frac{1}{d^2}$. Ceci est vrai s'il n'y a pas de réflexion. S'il y en a une, on considère que l'ordre de grandeur de cette relation reste vrai. On veut donc que $S_p \propto \frac{1}{d^2}$. On va définir en réalité deux seuils de puissance : le premier, S_p^t pour la phase d'entraînement de MLPd et le deuxième, S_p^v , pour la phase de validation de MLPd. En prenant $S_p^t < S_p^v$, on obtient de meilleurs résultats pour nos simulations. S_p^t et S_p^v dépendent aussi de la valeur du bruit, on les définit donc pour un RSB donné. En particulier, pour un RSB de 20 dB, on choisit empiriquement comme valeurs :

$$S_p^t = 0.01 \cdot \left(\frac{l}{d}\right)^2 \quad (3.29a)$$

$$S_p^v = 0.02 \cdot \left(\frac{l}{d}\right)^2 \quad (3.29b)$$

l est une constante de normalisation représentant une distance de 1 m. Avec de tels seuils, on garde tout de même la majorité des signaux, sauf ceux correspondant à des angles θ proches de 90° . Le bruit associé à ces signaux est trop grand, puisqu'il devient du même ordre de grandeur que la puissance moyenne P_m .

3.7.2 Optimisations de MLPd

Quand on utilise des réseaux de neurones, le choix des points d'entraînement est primordial, au niveau des entrées comme des sorties. Il est souvent plus judicieux d'avoir recours à des valeurs positives ou négatives quand on veut réaliser une classification binaire, comme c'est le cas ici : plutôt que d'utiliser les états 0 et 1, il est préférable de prendre les états -1 (hors du cône d'intérêt) et 1 (dans le cône d'intérêt). Les améliorations peuvent être considérables, comme on va le vérifier maintenant.

On compare deux structures dans les mêmes conditions : sorties 0 et 1 d'une part, sorties -1 et 1 d'autre part. Le RSB vaut 20 dB, on ne fait pas de pré-traitement, la réflexion parasite n'intervient pas (ni dans la phase d'entraînement, ni dans la phase de validation). Le maillage initial servant à prendre les DOAs dans le demi-espace $z > 0$ se compose de 50×50 points. On entraîne les deux MLPds pendant 30 époques et ils comportent chacun 10 neurones sur leur couche cachée.

Pour comparer les performances de ces deux RNAs, on définit un taux d'erreur T_e . Une erreur de classification est soit une DOA estimée dans le cône d'intérêt alors qu'elle n'y est pas (DOA à 0 ou -1 en réalité et estimée à 1 par MLPd), soit une DOA estimée hors du cône d'intérêt alors qu'elle y appartient effectivement (DOA à 1 en réalité et estimée à 0 ou -1 par MLPd). On appelle alors $N_{e,c}$ le nombre total d'erreurs de classification et $N_{d,tot}$ le nombre total de DOAs classées par MLPd (DOAs dont le signal est supérieur S_p).

$$T_e = 100 \frac{N_{e,c}}{N_{d,tot}} \quad (3.30)$$

Le Tableau 3.5 expose les performances des deux MLPs dans les conditions de simulation présentées précédemment.

	T_e
Sorties 0 et 1	19.15 %
Sorties -1 et 1	8.95 %

TAB. 3.5 *Comparaison de performances des deux MLPds*

Dorénavant, on va donc adopter les sorties -1 et 1 comme états possibles pour l'étage de détection. Le taux d'erreur est amélioré d'un facteur supérieur à 2. Cette différence de résultats très nette s'explique de cette manière : le temps d'apprentissage d'un RNA est nettement amélioré quand on utilise des signaux de moyenne nulle (Haykin, 1999). En effet, le temps d'apprentissage d'un RNA est de l'ordre de $\frac{\lambda_{max}}{\lambda_{min}}$, avec λ_{max} la valeur propre la plus grande du Hessien H et λ_{min} sa valeur propre non nulle la plus petite (voir Section 2.4.3). Or, les observations montrent que ce ratio est plus grand si les entrées des neurones sont de moyenne non nulle. Dans la mesure où on ne modifie pas nos vecteurs d'entrée (signaux de puissance de moyenne strictement positive donc), on peut améliorer la vitesse d'entraînement en adoptant des fonctions de transfert impaires. On utilise des sigmoïdes (fonctions impaires) pour les neurones situés dans la couche cachée de MLPd. Une augmentation de la vitesse d'apprentissage peut être réalisée si on utilise une fonction impaire pour le neurone de sortie. C'est le cas en prenant les sorties -1 et 1 plutôt que les sorties 0 et 1.

La prochaine étape de l'optimisation de MLPd consiste à déterminer les nombres optimaux de neurones sur sa couche cachée et d'époques d'entraînement. Les points d'entraînement sont équirépartis sur la demi-sphère unité d'axe Z de révolution. Un bruit de RSB égal à 20 dB est rajouté. Pour la phase de validation, on prend des points différents de ceux utilisés pour l'entraînement, mais également équirépartis sur la même demi-sphère. Le bruit rajouté est de RSB égal à 20 dB et une réflexion

parasite intervient, générée par le sol (route en asphalte, $\epsilon_r = 3.2$).

A nombre d'époques d'entraînement fixé (à 30), on trouve le nombre optimal de neurones sur la couche cachée (40). On cherche ensuite le nombre optimal d'époques d'entraînement pour un MLPd à 40 neurones sur sa couche cachée. On trouve que 45 époques est la solution la plus efficace. Cette méthode est beaucoup plus rapide que de réaliser une recherche exhaustive et permet d'aboutir au même couple (N_n^*, N_e^*) . Ceci est détaillé dans la Section 4.1. Le Tableau 3.6 expose les performances de MLPd optimal sous différents points de validation.

	T_e
RSB=20 dB, $N_{pre}=1$, sans réflexion	4.85 %
RSB=20 dB, $N_{pre}=30$, sans réflexion	1.31 %
RSB=20 dB, $N_{pre}=1$, avec réflexion	5.54 %
RSB=20 dB, $N_{pre}=30$, avec réflexion	1.36 %

TAB. 3.6 *Performances de MLPd optimal à 40 neurones entraîné pendant 45 époques*

On vérifie que des conditions de validation sous un RSB de 35 dB sans pré-traitement donnent les mêmes résultats que sous un RSB de 20 dB avec un pré-traitement effectué sur 30 échantillons. L'effet de la réflexion sur la qualité de la détection semble négligeable devant celui du bruit. Ceci est vrai pour une route ayant un $\epsilon_r = 3.2$. Sur route mouillée, ϵ_r augmente ($\epsilon_r = 80$ pour l'eau). En prenant $\epsilon_r = 15$, la détection devient plus difficile pour MLPd : $T_e = 2.00$ % dans ce cas avec un RSB de 35 dB, sans pré-traitement.

3.7.3 Figures des erreurs de classification de MLPd optimal

On représente les DOAs de validation, celles qui sont bien classées, ainsi que celles qui ont été mal classées, dans deux situations différentes : sous un RSB de 20 dB, sans pré-traitement et sans réflexion d'une part (pour voir l'effet du bruit), et sous un RSB de 35 dB, sans pré-traitement et avec une réflexion générée par une route vérifiant $\epsilon_r = 15$ d'autre part (pour voir l'effet de la réflexion). On prend une valeur ϵ_r différente de celle de l'asphalte (on prend celle du sol) pour avoir une réflexion d'amplitude plus élevée en moyenne et mieux apprécier la qualité de classification de MLPd en présence d'une réflexion parasite. Ce sont les Figures 3.19 et 3.20 respectivement.

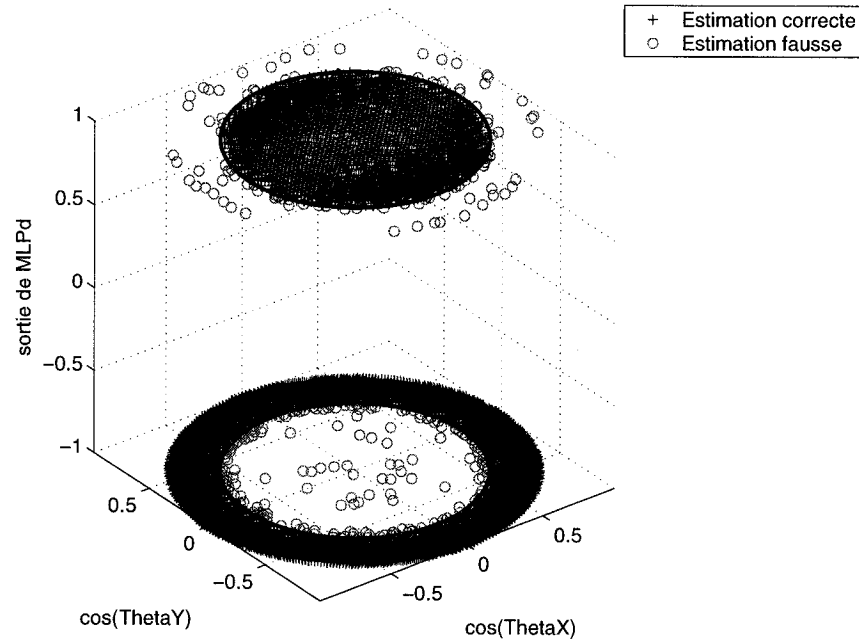


FIG. 3.19 Erreurs de classification de MLPd sous un RSB = 20 dB, sans réflexion

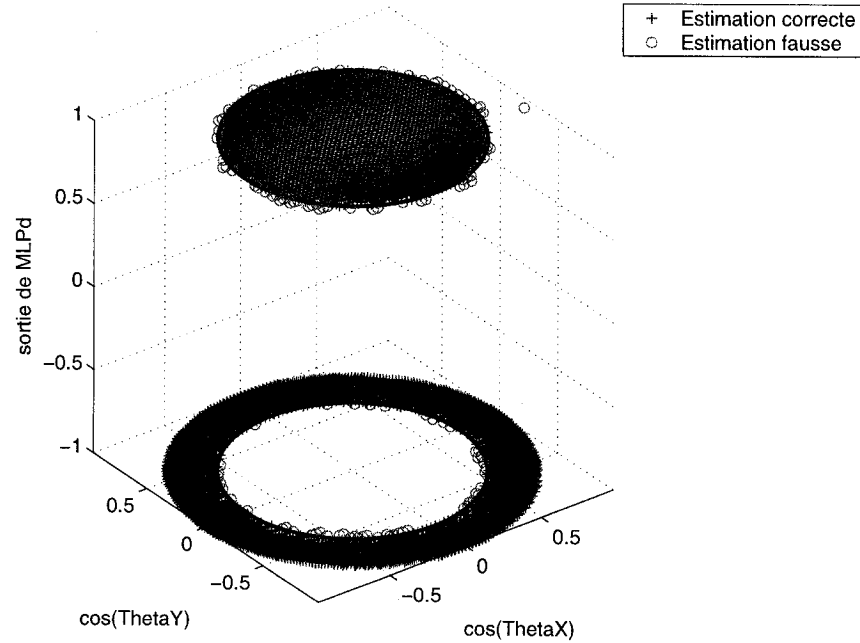


FIG. 3.20 Erreurs de classification de MLPd sous un $RSB = 35$ dB, avec réflexion, $\epsilon_r = 15$

Comme on peut le constater dans la Figure 3.19, MLPd fait des erreurs pour des DOAs qui ne sont pas seulement situées au niveau de la frontière du cône d'intérêt lorsque le bruit est important (20 dB). Les erreurs de classification qui changent un "1" en "-1" (faux négatif) ne sont pas importantes puisqu'elles signifient que l'étage d'estimation de la DOA ne sera pas activé. Le système attend le prochain vecteur d'entrée qui lui, sera peut-être moins bruité. Il faut surveiller par contre les erreurs changeant un "-1" en "1" (faux positif) : MLPx et MLPy vont essayer de retrouver une DOA qui n'appartient pas au cône d'intérêt. L'erreur angulaire sera d'autant plus importante que la DOA du signal est éloignée de la frontière du cône d'intérêt. Sur la Figure 3.20, il n'y a qu'une seule erreur de ce type très éloignée de la frontière sur l'ensemble des échantillons (plus de 7800, obtenus à partir d'un maillage 100×100), ce qui est excellent.

3.8 Conclusion

Dans ce chapitre, on a vu comment on a modélisé une réflexion parasite générée par une route. Ce modèle nous a permis d'optimiser notre système à plusieurs niveaux :

- on a pu déterminer le nombre optimal d'antennes pour chaque axe (trois plutôt que deux) et la polarisation la plus adaptée, i.e. une polarisation verticale.
- on a vu l'intérêt d'utiliser deux RNAs d'estimation différents, MLP_x et MLP_y, plutôt qu'un seul prenant les 16 signaux de puissance en entrée.

On a aussi décrit l'entraînement de MLP_d. On l'a testé en présence de bruit et d'une réflexion parasite pour déterminer ses performances.

Dans le prochain chapitre, on s'intéresse au choix des points d'entraînement pour encore réduire l'erreur angulaire d'estimation.

CHAPITRE 4

CHOIX DES POINTS D'ENTRAÎNEMENT DE MLPX ET MLPY

On a vu dans le chapitre précédent comment on a amélioré notre système pour faire face à une réflexion parasite. Etant donnée la structure neuronale que l'on s'est fixée (MLPs à une couche cachée), la prochaine étape consiste à réduire encore davantage les erreurs d'estimation, mais en jouant cette fois sur le choix des points d'entraînement. Voici les diverses méthodes qui ont été envisagées :

- l'entraînement “sans réflexion” (dit “entraînement D”), qui est l'entraînement classique généré avec un seul signal et du bruit ;
- l'entraînement “avec réflexion mono-configuration” (dit “entraînement DR”), utilisant un signal principal et sa réflexion générée par le modèle de réflexion sur route, avec d et ϵ_r constants ;
- l'entraînement “avec réflexion multi-configuration” (dit “entraînement DRM”), qui est une amélioration de l'entraînement DR puisque d et ϵ_r sont aléatoires ;
- l'entraînement “bi-signal classique” (dit “entraînement SS”), ayant recours à deux signaux indépendants et cohérents de DOAs fixes ;
- l'entraînement “bi-signal décalé” (dit “entraînement SSD”), basé sur deux signaux indépendants et cohérents de DOAs mobiles.

Dorénavant, la structure adoptée est composée de deux paires de réseaux orthogonaux de trois antennes, dont les éléments sont des antennes à polarisation verticale ayant un gain en $\cos\theta$. La fréquence de fonctionnement du dispositif est fixée à 77 GHz durant nos simulations, qui est souvent la fréquence utilisée pour les radars anti-collision équipant certains véhicules. On a malgré tout testé des valeurs de fréquence différentes (par exemple 5.8 GHz, fréquence qui avait été adoptée

dans (Coudyser, 2003) et (Fonseca, 2003)), ce qui ne change pas les résultats de simulation, notre système de détection étant construit en fonction de la longueur d’onde et donc de la fréquence (et non en distances absolues). On entraîne MLP_x et MLP_y sous un RSB de 20 dB, alors que les points de validation sont générés sous un RSB de 35 dB (équivalent à un RSB de 20 dB et un pré-traitement réalisé sur 30 échantillons). La réflexion est celle générée par une route en asphalte ou par le sol (de $\epsilon_r = 3.2$ et $\epsilon_r = 15$ respectivement) quand on place le centre R du système à une hauteur $h = 1\text{ m}$ du sol et l’émetteur E à une distance $d = 10\text{ m}$ de R (cas par défaut si aucune précision n’est mentionnée).

On teste deux valeurs différentes de ϵ_r pour étudier comment réagissent les divers entraînements neuronaux à des réflexions d’intensité différentes pour un angle d’incidence donné. Des routes en asphalte ou de terre sont courantes en pratique, c’est pourquoi ces deux valeurs particulières ont été testées.

4.1 Entraînement “sans réflexion” (D)

Cet entraînement est celui décrit à la fin du chapitre 2 : on utilise un maillage optimal du cône d’intérêt à partir d’un quadrillage carré pour obtenir les points d’entraînement (générés en présence de bruit) et de validation. La portion sphérique est tronquée (présence du sol) pour la phase de validation. On récapitule les résultats obtenus dans le Tableau 4.1.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}
30 N, 30 E	0.39°	1.17°	1.32°	1.71°	1.41°

TAB. 4.1 *Performances de l’entraînement sans réflexion testé sur MLP_x et MLP_y avec 30 neurones (N), entraînés pendant 30 époques (E)*

On a utilisé un réseau de 30 neurones entraîné pendant 30 époques pour MLPx et MLPy. On peut améliorer quelque peu cette configuration en déterminant le nombre optimal de neurones pour MLPx et MLPy entraînés sur 30 époques, puis en déterminant le nombre optimal d'époques d'entraînement pour le nombre optimal de neurones sur la couche cachée trouvé pour chacun d'entre eux.

Ce processus n'est pas censé trouver nécessairement le minimum global de l'erreur tracée en fonction de N_n et N_e (les nombres de neurones dans la couche cachée et d'époques d'entraînement respectivement). Pour le déterminer, il suffirait de tester tous les couples (N_n, N_e) par exemple et d'adopter le réseau donnant l'erreur minimale. Si on teste ces couples par incréments de 5 selon les deux directions, entre 10 et 50, cela revient à tester 81 couples différents, ce qui est beaucoup compte-tenu du temps moyen pour entraîner un seul RNA (autour de 15 min pour une structure moyenne entraînée à l'aide d'un PC performant d'un Go de RAM équipé d'un Pentium IV). Avec la pratique, on se rend compte que $N_n^* = 30$ pour la plupart des cas avec notre nombre de points d'entraînement (autour de 30000) : l'algorithme d'entraînement utilisé ("trainlm") a déjà convergé au bout de 30 époques. En extrapolant ainsi N_n^* , il ne nous reste plus qu'à tester 9 valeurs différentes de N_e pour arriver à N_e^* . On teste ensuite différentes valeurs de N_n sur des neurones ayant N_e^* neurones dans leur couche cachée. On vérifie bien ensuite que $N_n^* = 30$ dans la plupart des cas. On réduit ainsi le nombre de structures à tester à 18.

On a comparé ces deux méthodes sur divers cas et on a vérifié qu'on aboutissait au même couple (N_n^*, N_e^*) . Il est toutefois important de remarquer que de nombreux couples (N_n, N_e) donnent des résultats presque aussi bons que (N_n^*, N_e^*) , la différence de performance étant de l'ordre du dixième de degré sur l'estimation de θ_x ou θ_y . C'est pour cette raison que notre étude s'est plutôt orientée vers la comparaison de diverses méthodes d'entraînement, permettant d'améliorer les per-

formances angulaires de manière bien plus significative.

Finalement, la structure neuronale optimale dans le cadre de l'entraînement D est obtenue de la manière suivante :

- on entraîne un MLP composé de 40 neurones sur sa couche caché pendant 30 époques pour obtenir MLPx optimal,
- on entraîne un MLP composé de 50 neurones sur sa couche caché pendant 30 époques pour obtenir MLPy optimal.

Le Tableau 4.2 montre les résultats que l'on obtient avec cette nouvelle structure, avec deux valeurs différentes pour la permittivité relative : $\epsilon_r = 3.2$ et $\epsilon_r = 15$. On a rajouté une donnée supplémentaire dans la dernière case, l'écart-type moyen $\sigma_{av,y}$ calculé pour les estimations de l'angle θ_y . Il permet de caractériser les variations des données estimées pour cet angle. On le calcule à partir de l'écart-type $\bar{\sigma}_y(n)$ estimé pour la n^{ieme} tranche angulaire de 0.5° . Les croix de la Figure 4.3 par exemple, sont les sorties de MLPy moyennées sur ces tranches angulaires, et sont notées $\theta_{av,y}(n)$. On appelle $N_{abs}(n)$ le nombre de points présents dans la n^{ieme} tranche angulaire de 0.5° . Ce nombre est fonction de l'azimut et de l'élévation de la DOA. $\tilde{\theta}_y$ est toujours la sortie estimée par MLPy.

$$\bar{\sigma}_y(n) = \sqrt{\frac{1}{N_{abs}(n)} \sum_{j=1}^{N_{abs}(n)} (\tilde{\theta}_y(j) - \theta_{av,y}(n))^2} \quad (4.1)$$

On obtient alors facilement la relation entre $\bar{\sigma}_y(n)$ et $\sigma_{av,y}$. Si on appelle N_{tr} le nombre de tranches angulaires de 0.5° , on a :

$$\sigma_{av,y} = \frac{1}{N_{tr}} \sum_{j=1}^{N_{tr}} \bar{\sigma}_y(j) \quad (4.2)$$

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}	$\sigma_{av,y}$
Ent. D, $\epsilon_r = 3.2$	0.32°	1.05°	1.17°	1.55°	1.26°	0.65°
Ent. D, $\epsilon_r = 15$	0.38°	1.56°	1.70°	2.21°	1.56°	0.94°

TAB. 4.2 Performances de l'entraînement D testé sur MLPx et MLPy avec 40 et 50 neurones respectivement, entraînés pendant 30 époques, avec $\epsilon_r = 3.2$ et $\epsilon_r = 15$

La Figure 4.1 montre les points d'estimations réalisées par MLPx et MLPy dans le cadre de l'entraînement D, avec $\epsilon_r = 3.2$. La Figure 4.2 montre les estimations moyennées de MLPx, toujours avec $\epsilon_r = 3.2$. Prendre $\epsilon_r = 15$ ne change pas l'allure des résultats de MLPx. Les Figures 4.3 et 4.4 montrent les estimations moyennées de MLPy, avec l'erreur absolue d'estimation et l'écart-type respectivement, sous $\epsilon_r = 3.2$, tandis que les Figures 4.5 et 4.6 montrent les mêmes résultats sous $\epsilon_r = 15$.

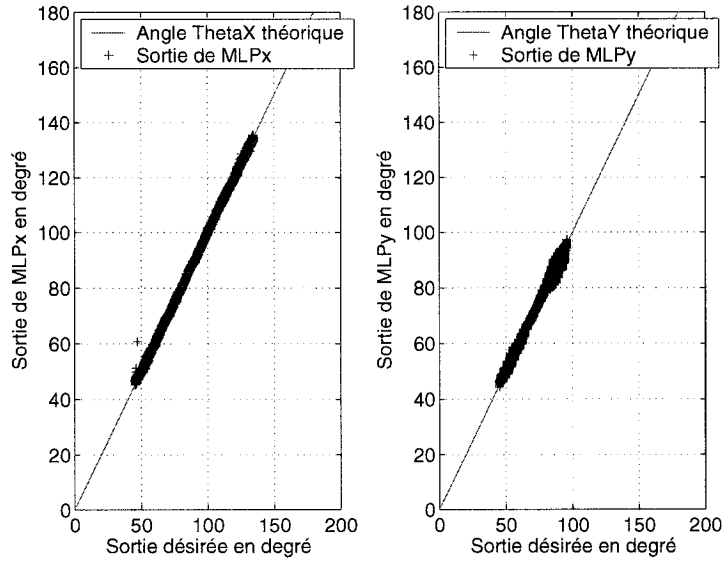


FIG. 4.1 Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite avec entraînement D et $\epsilon_r = 3.2$

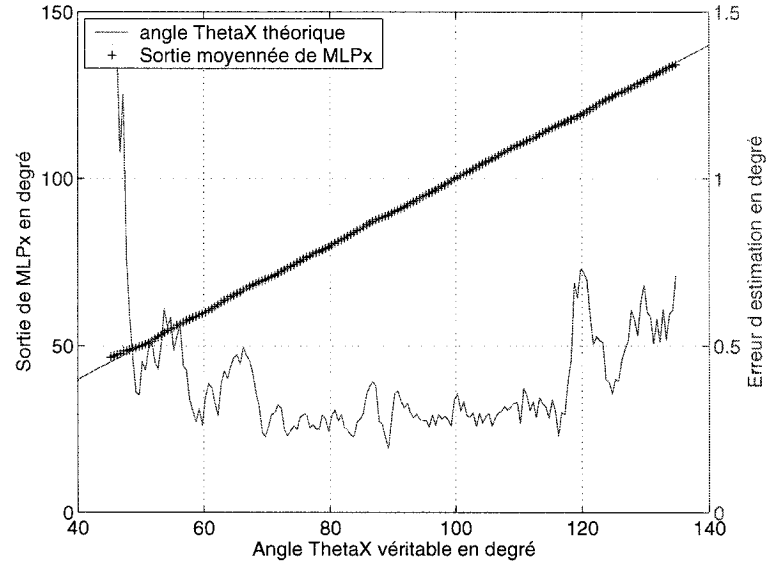


FIG. 4.2 Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx avec entraînement D et $\epsilon_r = 3.2$

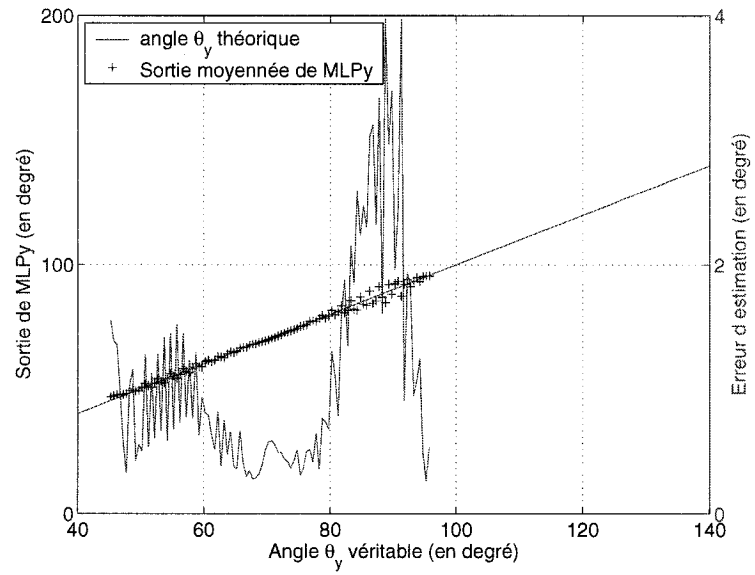


FIG. 4.3 Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec entraînement D et $\epsilon_r = 3.2$

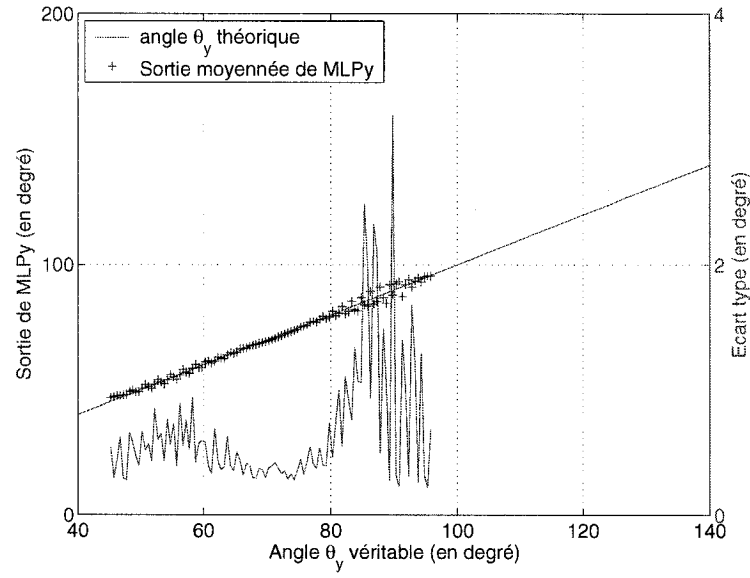


FIG. 4.4 *Sorties moyennées et écart-type des estimations de θ_y par MLPy avec entraînement D et $\epsilon_r = 3.2$*

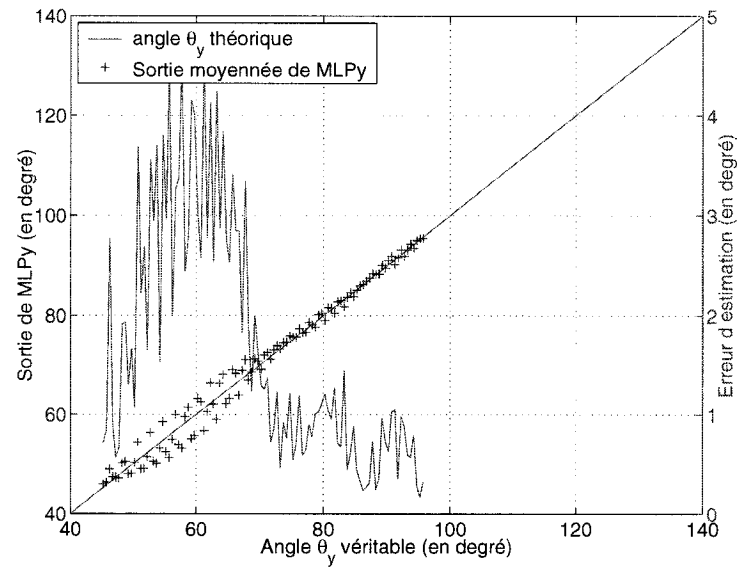


FIG. 4.5 *Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec entraînement D et $\epsilon_r = 15$*

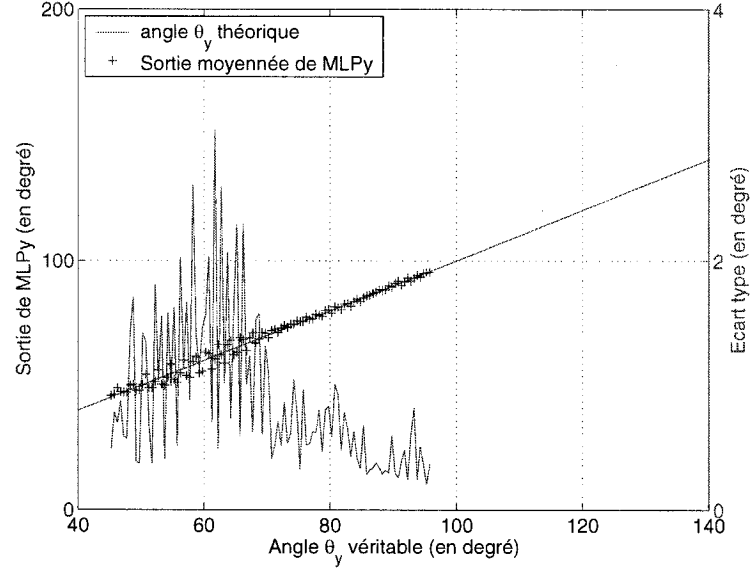


FIG. 4.6 *Sorties moyennées et écart-type des estimations de θ_y par MLPy avec entraînement D et $\epsilon_r = 15$*

L'amélioration provenant du choix d'une structure neuronale optimale par rapport à la structure de base (30 Neurones, 30 époques) n'est pas flagrante. Notre recherche va donc se focaliser sur de nouvelles méthodes d'entraînement.

4.2 Entraînement “avec réflexion mono-configuration” (DR)

Ici, on décide d'entraîner nos MLPs avec des signaux perturbés par du bruit et, nouveauté, une réflexion parasite, la même que celle utilisée pour la génération des points d'entraînement. Nos réseaux de neurones apprenant en quelque sorte l'effet d'une réflexion, on espère ainsi réduire l'erreur angulaire moyenne d'estimation, pour MLPy surtout. On place donc notre récepteur à une distance $h = 1\text{ m}$ du sol, avec une distance $d = 10\text{ m}$, sous un RSB de 20 dB. On teste ensuite notre système

avec des points de validation différents des points d'entraînement (maillage initial des DOAs différents et niveau de bruit différent : RSB de 35 dB). La génération de la réflexion se fait exactement de la même façon dans les deux cas (plan réflecteur identique). On commence avec MLPx et MLPy, RNAs comprenant 30 neurones sur leur couche cachée et entraînés sur 30 époques. Cette structure est très probablement proche de la structure finale optimale.

Le Tableau 4.3 expose les résultats des simulations avec l'entraînement DR.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}
Ent. DR	0.65°	0.98°	1.33°	1.74°	1.37°
gain p/r à Ent. D	-0.33°	+0.07°	-0.16°	-0.19°	-0.11°

TAB. 4.3 *Performances de l'entraînement DR testé sur MLPx et MLPy avec 30 neurones, entraînés pendant 30 époques*

On a réussi à diminuer l'erreur d'estimation de θ_y , mais au prix d'une détérioration sur celle de θ_x . Une solution consisterait à prendre MLPx entraîné selon l'entraînement D et MLPy entraîné selon l'entraînement DR.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}
d différente	0.52°	1.25°	1.47°	1.84°	1.57°
gain p/r à Ent. D	-0.19°	-0.20°	-0.30°	-0.29°	-0.31°

TAB. 4.4 *Performances de l'entraînement DR testé sur MLPx et MLPy avec 30 neurones, entraînés pendant 30 époques (d différente)*

Regardons ce qu'il se passe lorsque la distance d n'est plus la même entre l'entraînement et la validation. Pour cela, on entraîne MLPx et MLPy (30 neurones,

30 époques) avec une réflexion sur une route en asphalte, sous la condition $d = \sqrt{2}m$ (avec une telle distance, la sphère des DOAs n'est pas tronquée). La phase de validation se fait toujours avec $d = 10 m$. Le Tableau 4.4 montre ce que l'on obtient.

On assiste à une détérioration des résultats lorsque la distance d n'est pas la même entre les phases d'entraînement et de validation. Comme en pratique, cette distance n'est pas connue à l'avance (elle varie constamment), cet entraînement est moins bon que l'entraînement D.

Une remarque similaire concerne le fait que l'on ne connaît pas non plus à l'avance les propriétés électriques du matériau réflecteur. Surtout, sa permittivité relative ϵ_r n'est pas connue de façon précise, et elle est non uniforme. On vérifie l'importance de ce paramètre en entraînant les mêmes neurones (30 neurones, 30 époques), avec $d = 10 m$ pendant les phases d'entraînement et de validation. Par contre, $\epsilon_r = 3.2$ pendant la phase d'entraînement et $\epsilon_r = 15$ pendant la phase de validation. Le Tableau 4.5 montre les différences de résultats.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}
ϵ_r différente	0.66°	1.64°	1.92°	2.30°	1.84°
gain p/r à Ent. D	-0.28°	-0.08°	-0.22°	-0.09°	-0.28°

TAB. 4.5 *Performances de l'entraînement DR testé sur MLPx et MLPy avec 30 neurones, entraînés pendant 30 époques (ϵ_r différente)*

Là encore, on trouve que l'entraînement est moins bon que l'entraînement D en terme de minimisation des différentes erreurs angulaires définies. La solution consiste peut-être à faire varier ces paramètres pendant la phase d'entraînement, de manière à modéliser les différentes situations rencontrées en pratique.

4.3 Entraînement “avec réflexion multi-configuration” (DRM)

On entraîne toujours nos MLPs avec une réflexion, mais cette fois, on essaie de ne pas se limiter à un cas particulier de réflexion pendant l’entraînement. Au contraire, on essaie de généraliser au maximum les cas de figure possibles en modifiant les caractéristiques de la réflexion d’une DOA à l’autre. Ainsi, la distance d et la permittivité relative ϵ_r du matériau réflecteur sont prises aléatoires (on verra selon quelles lois) pour toutes les DOAs. On espère de cette façon faire apprendre à notre réseau les différentes situations que l’on peut rencontrer en pratique.

Pendant la phase d’entraînement, ϵ_r est une variable aléatoire indépendante de la DOA et des autres paramètres. Elle peut prendre des valeurs comprises entre 1 et 20, de manière équiprobable. En fait, tout se passe comme si on changeait le matériau réflecteur à chaque point d’entraînement. On fait également varier de manière aléatoire la distance d entre chaque DOA d’entraînement. On appelle Ω l’angle entre l’horizontale et la demi-droite définissant la limite inférieure du cône d’intérêt dans le plan $\theta_x = 90^\circ$. Cet angle varie avec la distance d à cause de la présence de la route (on enlève les rayons souterrains impossibles physiquement). La Figure 4.7 représente les intersections de deux cônes différents (pour $d = d1$ et $d = d2$) avec le plan $\theta_x = 90^\circ$. La demi-droite définissant la limite supérieure du cône est tout le temps la même par contre.

L’équation 4.3 donne la relation entre d et Ω .

$$d = \frac{h}{\sin \Omega} \quad (4.3)$$

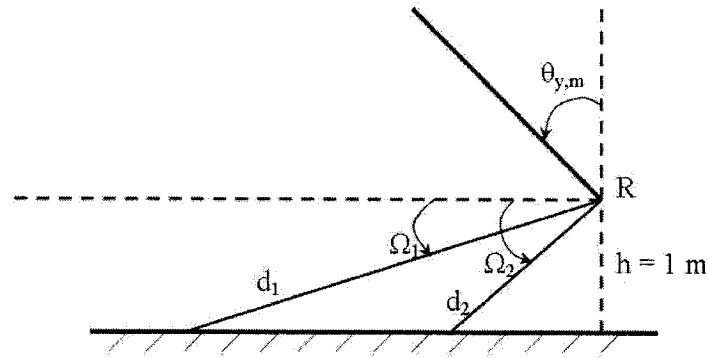


FIG. 4.7 Limites de deux cônes d'intérêt différents avec le plan $\theta_x = 90^\circ$

Pour générer des distances d aléatoires entre chaque DOA, on définit Ω comme étant une variable aléatoire uniformément distribuée sur l'intervalle $[0^\circ, 45^\circ]$. A chaque échantillon correspond une certaine valeur pour d . Cette méthode pour générer des distances d aléatoires est bonne dans la mesure où elle ne privilégie pas d'ouverture angulaire : on n'a pas de sur-représentation de cas où l'émetteur est éloigné ou au contraire proche du récepteur. Le Tableau 4.6 montre quelques valeurs numériques du couple (Ω, d) .

Ω en degré	d en mètre
45°	1.414 m
33.75°	1.800 m
22.5°	2.613 m
11.25°	5.126 m
0.573°	100 m

TAB. 4.6 Valeurs numériques de Ω et d

La Figure 4.8 représente la fonction de densité de probabilité de la distance d selon cette méthode.

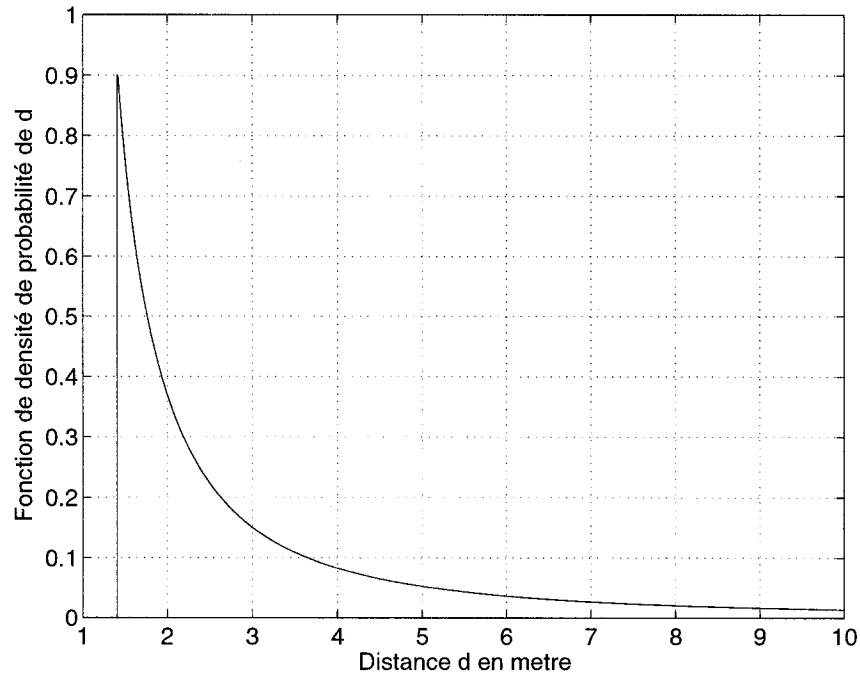


FIG. 4.8 *Fonction de densité de probabilité de d*

Comme précédemment, ces nouveaux signaux servent à entraîner MLPx et MLPy, des RNAs ayant chacun 30 neurones sur leur couche cachée, pendant 30 époques. Les points de validation sont obtenus pour une distance $d = 10\text{ m}$ avec $\epsilon_r = 15$. On a ensuite cherché à optimiser MLPx et MLPy pour finalement aboutir aux mêmes conclusions que pour l'entraînement traditionnel, à savoir que MLPx optimal est un RNA à 40 neurones entraîné pendant 30 époques et que MLPy optimal est un RNA à 50 neurones entraîné pendant 30 époques aussi. Le Tableau 4.7 met en valeur les performances de ces deux systèmes.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}
MLPx et MLPy 30 N 30 E	0.90°	1.94°	2.36°	2.88°	2.20°
gain p/r à Ent. D	-0.52°	-0.38°	-0.66°	-0.67°	-0.64°
MLPx et MLPy optimaux	0.84°	1.67°	2.08°	2.48°	1.95°
gain p/r à Ent. D	-0.46°	-0.11°	-0.38°	-0.27°	-0.39°

TAB. 4.7 Performances de l'entraînement DRM testé sur $(MLPx, MLPy) = (30N, 30E, 30N, 30E)$ et $(MLPx, MLPy) = (40N, 30E, 50N, 30E)$

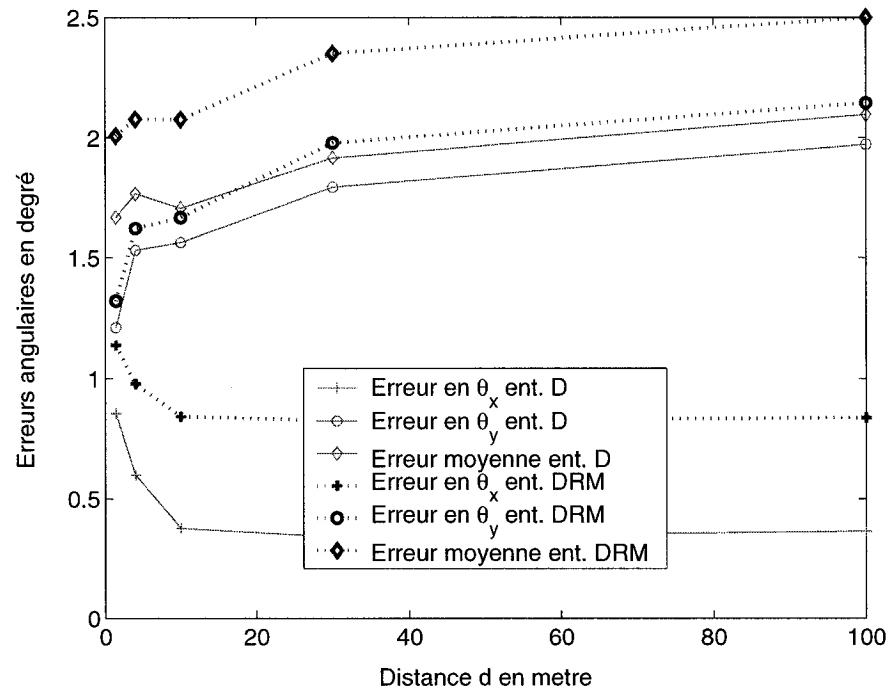


FIG. 4.9 Comparaison des performances de l'entraînement D et de l'entraînement DRM, en fonction de la distance d

On reste vraiment proche des résultats de l'entraînement DR. Pour savoir si l'entraînement DRM est meilleur ou non que l'entraînement D, on a testé les structures

optimales (avec l'entraînement D et l'entraînement DRM) pour des distances d diverses s'échelonnant de $\sqrt{2}$ à 100 m avec une permittivité relative du matériau réflecteur constante et égale à 15. Les résultats sont présentés sur la Figure 4.9

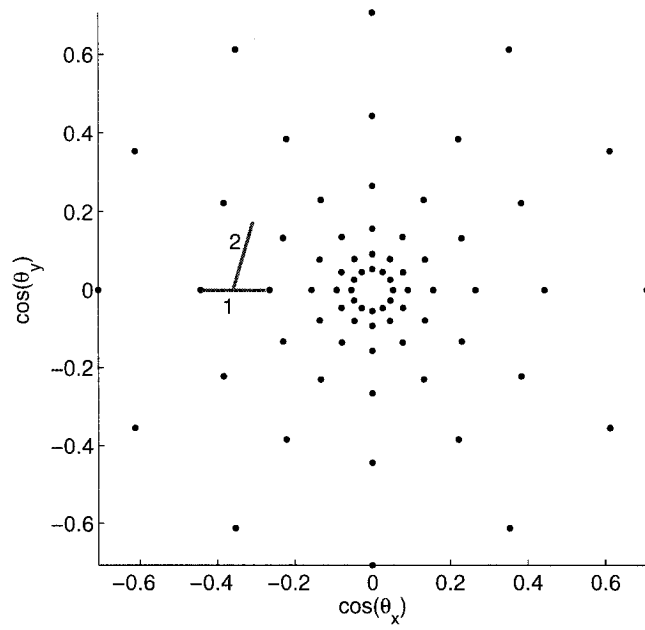
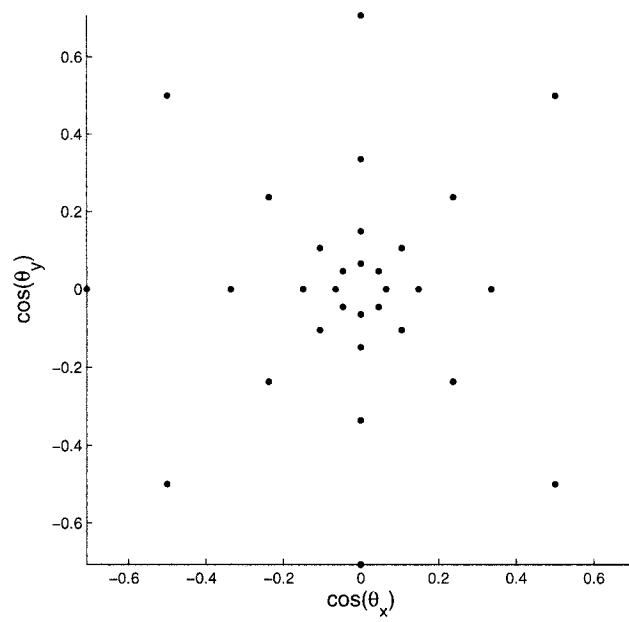
Il apparaît que l'entraînement DRM donne de moins bons résultats que l'entraînement D, du moins dans les simulations. On vérifiera cette assertion dans le dernier chapitre, avec des mesures expérimentales.

4.4 Entraînement “bi-signal classique” (SS)

Cette section décrit une nouvelle méthode de génération de points d'entraînement. Elle ne repose plus sur le modèle de réflexion sur route. Elle peut être appliquée à toute situation de DOA dans laquelle le système de réception reçoit deux signaux : le faisceau principal et une réflexion parasite, d'intensité moindre.

Le point de départ consiste à réaliser un échantillonnage de DOAs du cône d'intérêt. Pour cela, on définit la portion de sphère unité inscrite dans le cône d'intérêt. Tout point situé sur cette portion sphérique peut être considéré comme une DOA incluse dans le cône d'intérêt. On veut échantillonner deux portions sphériques différentes et indépendantes : la première indiquera les DOAs du signal direct, la deuxième celle de la réflexion.

La première étape réside dans le choix du type d'échantillonnage de la portion sphérique. Les deux caractéristiques primordiales sont : une bonne symétrie du choix des DOAs en azimut et élévation et un maillage plus dense près du pôle de la sphère (plus de DOAs le long de l'axe Z que sur le bord du cône). Cette deuxième propriété permettra d'épargner de la mémoire pour la phase d'entraînement et d'obtenir une meilleure précision pour des DOAs proches de l'axe principal de propagation.

FIG. 4.10 *DOAs du signal direct*FIG. 4.11 *DOAs du signal réfléchi*

Il est implicite que la probabilité de réception sera plus grande au centre du cône d'intérêt lors d'une utilisation future. Les Figures 4.10 et 4.11 représentent les DOAs adoptées pour le faisceau principal et la réflexion respectivement.

Pour réaliser un tel échantillonnage, on commence par se fixer un nombre de méridiens (équivalents à l'azimut). L'angle formé entre deux méridiens consécutifs est constant, donc deux points se déplaçant à la même vitesse le long de deux méridiens consécutifs s'éloignent de plus en plus. La densité des points sera donc plus grande près de l'axe Z comme désiré. Le prochain problème consiste à trouver l'espacement entre chaque parallèle (qui correspond à l'élévation) tel que l'on ait une bonne symétrie en azimut et élévation. Concrètement, ceci revient à égaliser les distances des deux segments représentés sur la Figure 4.10 quelle que soit leur position sur la carte.

On définit une DOA par un couple (θ, ϕ) correspondant à son élévation et son azimut respectivement. Le problème consiste donc à déterminer les deux suites θ_n et ϕ_m donnant le choix de ces DOAs. Se fixer un nombre de méridiens équidistants entre eux revient à dire que ϕ_m est une suite arithmétique définie par les équations 4.4.

$$\Delta_\phi = \frac{2\pi}{M+1} \quad (4.4a)$$

$$\phi_0 = 0 \quad (4.4b)$$

$$\phi_M = 2\pi - \Delta_\phi \quad (4.4c)$$

$$\phi_m = \phi_{m-1} + \Delta_\phi \quad (4.4d)$$

ϕ_m est une suite de $M+1$ éléments. $M+1 = 12$ pour la Figure 4.10 et $M+1 = 8$ pour la Figure 4.11. On prend plus de méridiens pour le signal direct que pour la réflexion puisque l'estimation de la DOA se fera sur le faisceau direct.

Il faut maintenant identifier la suite θ_n . On part de l'équation 4.5 traduisant la

recherche de symétrie entre l'azimut et l'élévation.

$$\frac{\theta_n + \theta_{n-1}}{2} \Delta_\phi = \theta_n - \theta_{n-1} \quad (4.5)$$

En effet, $\frac{\theta_n + \theta_{n-1}}{2} \Delta_\phi$ représente la longueur d'un arc de cercle. On considère que cette expression donne une bonne approximation de la longueur du segment 1 sur la Figure 4.10. $\theta_n - \theta_{n-1}$ représente la longueur exacte du segment 2 sur la même figure.

Finalement, on trouve que la suite θ_n est géométrique et est totalement définie par les équations 4.6.

$$\frac{\theta_n}{\theta_{n-1}} = \frac{1 + \frac{\Delta_\phi}{2}}{1 - \frac{\Delta_\phi}{2}} = \Gamma_\theta > 1 \quad (4.6a)$$

$$\theta_N = \frac{\pi}{4} \quad (4.6b)$$

$$\theta_0 = \frac{\theta_N}{(\Gamma_\theta)^N} \quad (4.6c)$$

θ_n est une suite de $N+1$ éléments. $N+1 = 6$ pour la Figure 4.10 et $N+1 = 4$ pour la Figure 4.11. En prenant $N+1 = \frac{M+1}{2}$, on s'assure un échantillonnage ni trop dense au centre, ni trop espacé sur les bords.

De cette façon, on prend 72 ($= 12 \cdot 6$) DOAs différentes pour le signal direct et 32 ($= 8 \cdot 4$) pour la réflexion. Pour chacune des DOAs du signal direct (ou DOA^d), on associe les 32 DOAs possibles de la réflexion (ou DOA^r). On génère ainsi 2304 ($= 72 \cdot 32$) configurations possibles, une configuration étant définie par un couple (DOA^d, DOA^r) .

Ensuite, on somme vectoriellement les signaux direct et réfléchi au niveau de chacune des 6 antennes, en pondérant la réflexion d'un coefficient positif $\Gamma_v < 1$. Ce

poids est censé représenter l'effet du coefficient de réflexion vertical C_v . Pour chacun des 2304 couples précédents, on choisit une certaine valeur pour Γ_v comprise entre 0 et 0.6. En effet, on a vu quantitativement que $C_v \in [0 ; 0.6]$ pour la plupart des cas rencontrés selon le modèle de réflexion sur route. Γ_v prend une des 13 valeurs du vecteur $[0 \ 0.05 \ 0.1 \ 0.15 \ 0.2 \ 0.25 \ 0.3 \ 0.35 \ 0.4 \ 0.45 \ 0.5 \ 0.55 \ 0.6]$. On génère ainsi 29952 ($= 2304 \cdot 13$) situations différentes. Les équations 4.7 expriment les signaux complexes en sortie de chacune des 6 antennes.

$$A_{x1,eb} = G(\theta) + \Gamma_v G(\theta_r) \quad (4.7a)$$

$$A_{x2,eb} = G(\theta) \exp(2\pi j \cdot 0.58 \cdot \cos \theta_x) + \Gamma_v G(\theta_r) \exp(2\pi j \cdot 0.58 \cdot \cos \theta_{x,r}) \quad (4.7b)$$

$$A_{x3,eb} = G(\theta) \exp(2\pi j \cdot 1.37 \cdot \cos \theta_x) + \Gamma_v G(\theta_r) \exp(2\pi j \cdot 1.37 \cdot \cos \theta_{x,r}) \quad (4.7c)$$

$$A_{y1,eb} = G(\theta) + \Gamma_v G(\theta_r) \quad (4.7d)$$

$$A_{y2,eb} = G(\theta) \exp(2\pi j \cdot 0.58 \cdot \cos \theta_y) + \Gamma_v G(\theta_r) \exp(2\pi j \cdot 0.58 \cdot \cos \theta_{y,r}) \quad (4.7e)$$

$$A_{y3,eb} = G(\theta) \exp(2\pi j \cdot 1.37 \cdot \cos \theta_y) + \Gamma_v G(\theta_r) \exp(2\pi j \cdot 1.37 \cdot \cos \theta_{y,r}) \quad (4.7f)$$

On les combine entre eux dans le combinateur de signaux, on rajoute du bruit, on les normalise pour alimenter les MLPs. On obtient finalement 29952 vecteurs de 16 composantes pour entraîner MLPd, MLPx et MLPy.

On entraîne MLPx et MLPy (avec 30 neurones chacun sur leur couche cachée) pendant 30 époques. On calcule ensuite les erreurs angulaires avec des points de validation obtenus à partir du modèle de réflexion sur route, exactement comme les cas précédents. On arrive de cette manière à déterminer le nombre optimal de neurones dans la couche cachée (50 pour MLPy et 30 pour MLPx), puis le nombre optimal d'époques d'entraînement (30 pour MLPx et MLPy). Le Tableau 4.8 montre les performances de la structure optimale avec un entraînement SS. On a pris $\epsilon_r = 15$, $h = 1 \text{ m}$ et deux valeurs pour d : 10 et 100 m.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}
$d = 10\text{ m}$	1.13°	2.11°	2.75°	3.51°	2.46°
gain p/r à Ent. D	-0.55°	-0.55°	-1.05°	-1.30°	-0.90°
$d = 100\text{ m}$	1.12°	2.51°	3.07°	3.83°	2.81°

TAB. 4.8 Performances de l'entraînement SS avec architecture optimisée pour $d = 10\text{ m}$ et $d = 100\text{ m}$

Cet entraînement est moins bon que l'entraînement D. Mais on peut l'améliorer grandement en déplaçant les DOAs en “bouchant les trous” des maillages tout en prenant la même place en mémoire. C'est là l'avantage de l'entraînement bi-signal décalé décrit dans la suite.

4.5 Entraînement “bi-signal décalé” (SSD)

Les DOAs des signaux direct et réfléchi étaient fixées dès le début pour l'entraînement bi-signal. En conséquence, de nombreuses DOAs de validation étaient souvent assez éloignées des DOAs d'entraînement, d'où une erreur angulaire d'estimation assez élevée. Pour une DOA^r , on associait le maillage des DOA^d . Quand on prenait une autre valeur de DOA^r , le maillage des DOA^d restait le même.

L'idée de base de l'entraînement SSD consiste à décaler l'ensemble du maillage des DOA^d quand on change de DOA^r . On décide également de décaler tout le maillage des DOA^r quand on change la valeur du coefficient Γ_v . De cette manière, pour tout point de validation, on s'assure de pouvoir trouver un point d'entraînement SSD bien plus proche de celui-ci que l'on pourrait avoir avec l'entraînement bi-signal classique (statique). L'erreur angulaire d'estimation s'en trouve donc réduite. La Figure 4.12 représente un exemple de décalage du maillage originel des DOAs sur

un nombre réduit de points (pour une meilleure visualisation).

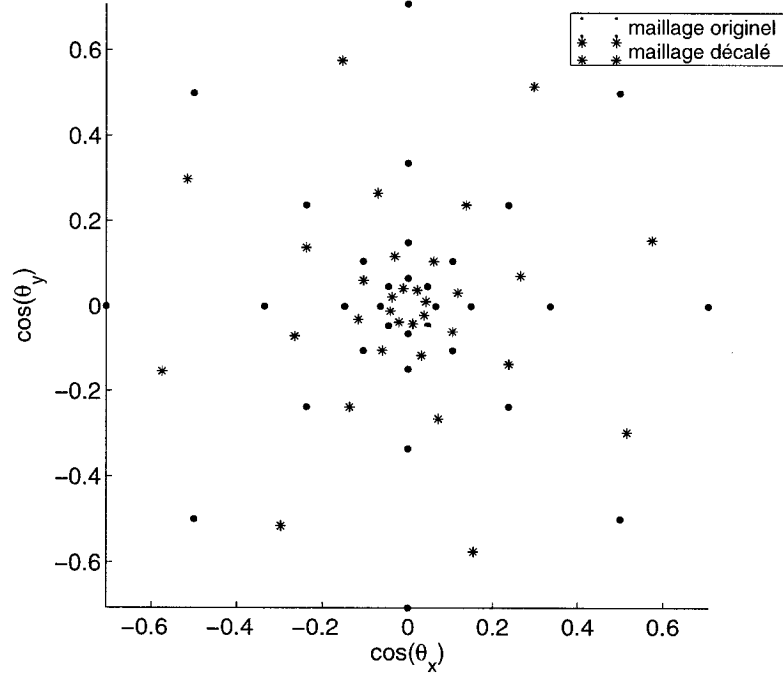


FIG. 4.12 Exemple de décalage des DOAs à la base de l'entraînement SSD

On définit entièrement un point d'entraînement SSD par la donnée d'un sextuplet $(\theta_d, \phi_d, \theta_r, \phi_r, k_1, k_2)$. θ_d et ϕ_d représentent les angles sphériques du signal direct, θ_r et ϕ_r représentent les angles sphériques de la réflexion, k_1 et k_2 sont les indices de la matrice $(\Gamma_v)_{k_1, k_2}$. Chaque composante de cette matrice est une valeur particulière (positive) du coefficient de réflexion. On définit $(N_d + 1) \cdot (M_d + 1) \cdot (N_r + 1) \cdot (M_r + 1) \cdot K_1 \cdot K_2 = N_{p, SSD}$ points d'entraînement différents. $(N_d + 1)$ et $(M_d + 1)$ sont respectivement les nombres de parallèles et de méridiens du maillage initial du signal direct, $(N_r + 1)$ et $(M_r + 1)$ sont respectivement les nombres de parallèles et de méridiens du maillage initial de la réflexion, K_1 et K_2 sont respectivement les nombres de lignes et de colonnes de la matrice $(\Gamma_v)_{k_1, k_2}$. Voyons quelles sont les va-

leurs du sextuplet pour le p^{ieme} point d'entraînement, p variant de 0 à $(N_{p,SSD} - 1)$. Pour cela, on décompose p de la façon décrite par les équations 4.8 (décomposition unique) :

$$p = n_1 K_2 (N_r + 1) (M_r + 1) (N_d + 1) (M_d + 1) + r_1 , \quad (4.8a)$$

$$r_1 < K_2 (N_r + 1) (M_r + 1) (N_d + 1) (M_d + 1)$$

$$r_1 = n_2 (N_r + 1) (M_r + 1) (N_d + 1) (M_d + 1) + r_2 , \quad (4.8b)$$

$$r_2 < (N_r + 1) (M_r + 1) (N_d + 1) (M_d + 1)$$

$$r_2 = n_r (M_r + 1) (N_d + 1) (M_d + 1) + r_3 , \quad r_3 < (M_r + 1) (N_d + 1) (M_d + 1) \quad (4.8c)$$

$$r_3 = m_r (N_d + 1) (M_d + 1) + r_4 , \quad r_4 < (N_d + 1) (M_d + 1) \quad (4.8d)$$

$$r_4 = n_d (M_d + 1) + m_d , \quad m_d < (M_d + 1) \quad (4.8e)$$

On peut maintenant expliciter les six composantes du sextuplet :

$$\phi_d = m_d \Delta_{\phi_d} + m_r \frac{\Delta_{\phi_d}}{M_r + 1} , \quad \Delta_{\phi_d} = \frac{2\pi}{M_d + 1} \quad (4.9a)$$

$$\theta_d = (\Gamma_{\theta_d})^{n_d} \theta_{0d} - \frac{n_r}{N_r + 1} \theta_{0d} [(\Gamma_{\theta_d})^{n_d} - (\Gamma_{\theta_d})^{n_d-1}] ,$$

$$\Gamma_{\theta_d} = \frac{1 + \frac{\Delta_{\phi_d}}{2}}{1 - \frac{\Delta_{\phi_d}}{2}} , \quad \theta_{0d} = \frac{\pi}{4(\Gamma_{\theta_d})^{N_d}} \quad (4.9b)$$

$$\phi_r = m_r \Delta_{\phi_r} + n_2 \frac{\Delta_{\phi_r}}{K_2} , \quad \Delta_{\phi_r} = \frac{2\pi}{M_r + 1} \quad (4.9c)$$

$$\theta_r = (\Gamma_{\theta_r})^{n_r} \theta_{0r} - \frac{n_1}{K_1} \theta_{0r} [(\Gamma_{\theta_r})^{n_r} - (\Gamma_{\theta_r})^{n_r-1}] ,$$

$$\Gamma_{\theta_r} = \frac{1 + \frac{\Delta_{\phi_r}}{2}}{1 - \frac{\Delta_{\phi_r}}{2}} , \quad \theta_{0r} = \frac{\pi}{4(\Gamma_{\theta_r})^{N_r}} \quad (4.9d)$$

$$k_2 = n_2 + 1 \quad (4.9e)$$

$$k_1 = n_1 + 1 \quad (4.9f)$$

Pour nos simulations, on s'est fixé $N_d + 1 = 6$, $M_d + 1 = 12$, $N_r + 1 = 4$, $M_r + 1 = 8$, $K_1 = 5$ et $K_2 = 3$, soit $N_{p,SSD} = 34560$. On donne également la matrice $(\Gamma_v)_{k_1,k_2}$ adoptée :

$$(\Gamma_v)_{k_1,k_2} = \begin{pmatrix} 0 & 0.56 & 0.04 \\ 0.52 & 0.08 & 0.48 \\ 0.12 & 0.44 & 0.16 \\ 0.40 & 0.20 & 0.36 \\ 0.24 & 0.32 & 0.28 \end{pmatrix}$$

En alternant les valeurs fortes et faibles de C_v dans la matrice, on obtient de meilleurs résultats. En effet, les points d'entraînement obtenus avec deux valeurs de C_v consécutives dans cette matrice (et formant deux jeux de points), sont très proches. Pour optimiser au mieux la richesse de nos vecteurs d'entraînement, on va chercher à éloigner le plus possible ces deux jeux de points en prenant deux valeurs de C_v très différentes.

Ayant ainsi construit notre matrice de points d'entraînement, on part d'une structure neuronale initiale de 30 neurones sur la couche cachée que l'on entraîne pendant 30 époques, pour MLPx et MLPy. Durant cet entraînement, on s'aperçoit qu'il est préférable d'utiliser un bruit d'un RSB de 25 dB plutôt que de 20 dB, on obtient en effet une erreur angulaire plus faible sur nos points de validation par la suite. Ceci s'explique par le fait qu'on a construit un jeu de points d'entraînement efficace pour déterminer les hyperplans optimaux sans avoir recours à un bruit élevé. On teste toujours la structure avec des points de validation entachés d'un bruit de RSB 35 dB. Finalement, on arrive à la conclusion que les structures optimales sont des RNAs de 40 et 30 neurones, entraînés pendant 40 époques chacun, pour MLPx et MLPy respectivement. Le Tableau 4.9 expose les résultats obtenus avec des distances $d = 10\ m$ et $h = 1\ m$.

	$\epsilon_{av,x}$	$\epsilon_{av,y}$	ϵ_{av}	ϵ_{ms}	ϵ_{mc}	$\sigma_{av,y}$
$\epsilon_r = 3.2$	0.55°	1.31°	1.56°	1.97°	1.57°	0.89°
gain p/r à Ent. D	-0.23°	-0.26°	-0.39°	-0.42°	-0.31°	-0.24°
$\epsilon_r = 15$	0.57°	1.46°	1.72°	2.23°	1.55°	1.04°
gain p/r à Ent. D	-0.19°	+0.10°	-0.02°	-0.02°	+0.01°	-0.10°

TAB. 4.9 Performances de l'entraînement SSD avec architecture optimisée pour $\epsilon_r = 3.2$ et $\epsilon_r = 15$

On remarque que pour $\epsilon_r = 15$, les résultats sont similaires à ceux obtenus avec l'entraînement D. Il est intéressant d'étudier l'allure des réponses angulaires de MLPx et MLPy, ainsi que leurs erreurs et écart-types, grâce aux Figures 4.13, 4.14, 4.15, 4.16, 4.17 et 4.18 obtenues pour $\epsilon_r = 3.2$ et $\epsilon_r = 15$.

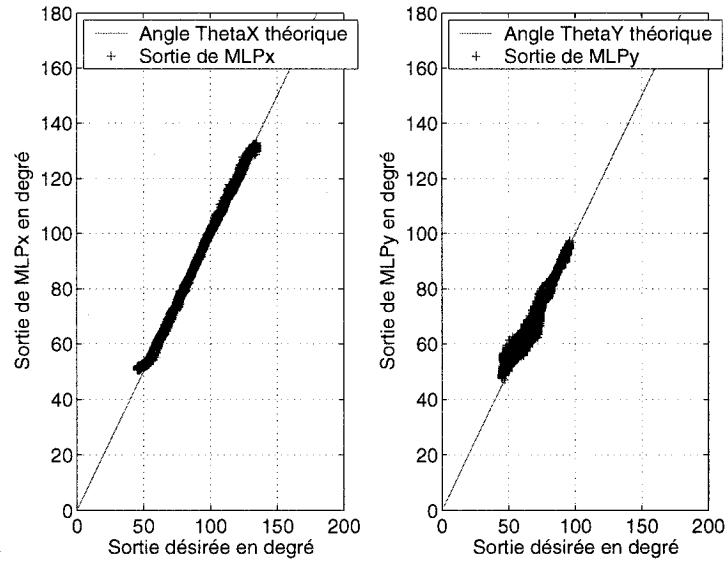


FIG. 4.13 Estimations de θ_x par MLPx à gauche et de θ_y par MLPy à droite avec entraînement SSD

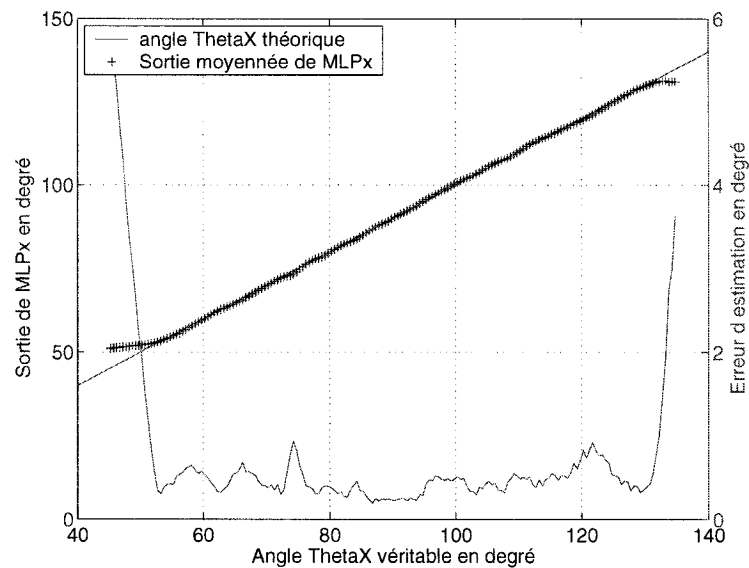


FIG. 4.14 *Sorties moyennées et erreur absolue d'estimation de θ_x par MLPx avec entraînement SSD*

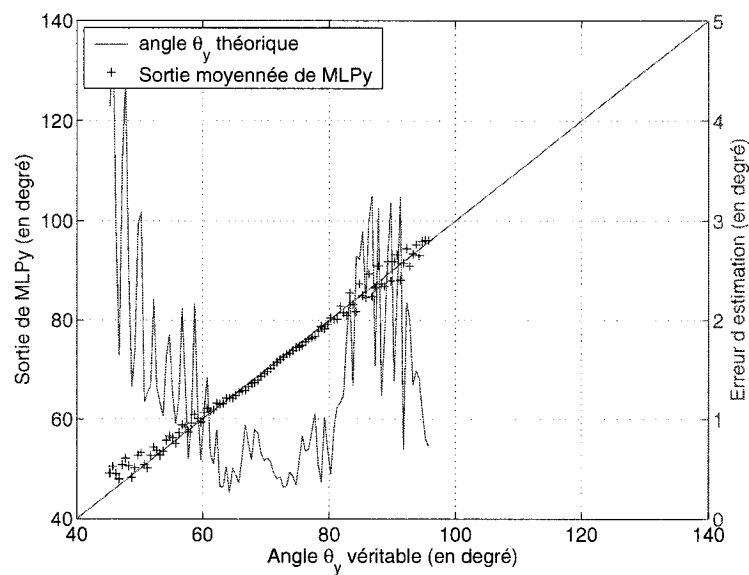


FIG. 4.15 *Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec entraînement SSD et $\epsilon_r = 3.2$*

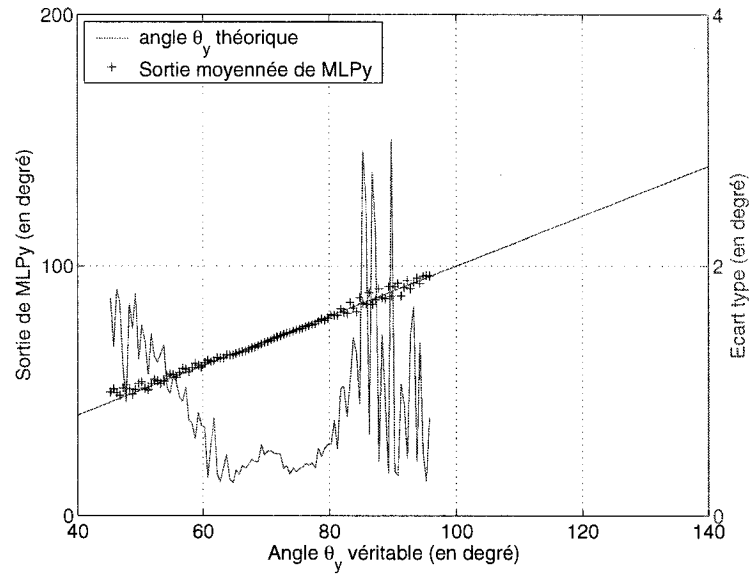


FIG. 4.16 *Sorties moyennées et écart-type des estimations de θ_y par MLPy avec entraînement SSD et $\epsilon_r = 3.2$*

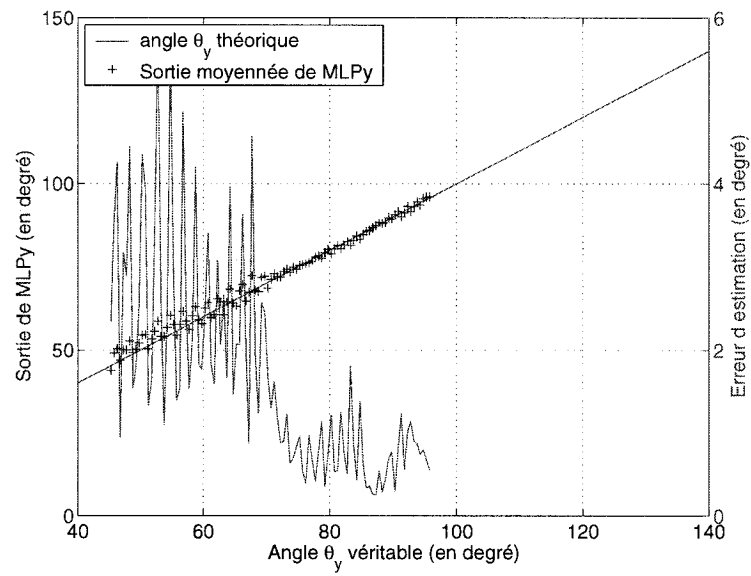


FIG. 4.17 *Sorties moyennées et erreur absolue d'estimation de θ_y par MLPy avec entraînement SSD et $\epsilon_r = 15$*

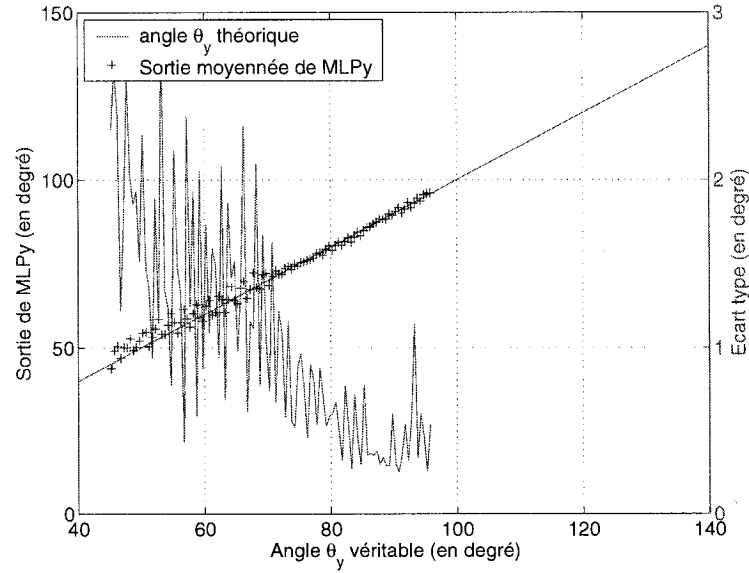


FIG. 4.18 *Sorties moyennées et écart-type des estimations de θ_y par MLPy avec entraînement SSD et $\epsilon_r = 15$*

En comparant ces courbes et leurs homologues obtenues avec l'entraînement D, on peut remarquer que l'entraînement SSD est meilleur que l'entraînement D pour les incidences rasantes situées autour de l'axe Z, quand $\epsilon_r = 3.2$, i.e. pour une route sèche en asphalte. En effet, les réflexions sont fortes pour des DOAs proches de la direction principale d'arrivée quand ϵ_r est faible (angle de Brewster décalé vers des angles d'élévation élevés d'après la Figure 3.3). L'entraînement SSD semble donc plus robuste que l'entraînement D pour des réflexions de forte intensité. Mais dans l'ensemble, l'entraînement D paraît donner de meilleures estimations d'après le Tableau 4.9, si on regarde les données $\epsilon_{av,y}$ et $\sigma_{av,y}$ obtenues avec ces deux entraînements.

Il est aussi intéressant de comparer les entraînements D et SSD pour des distances d différentes (utilisées pour générer les points de validation), comprises entre $\sqrt{2}$

et 100 m (Figure 4.19).

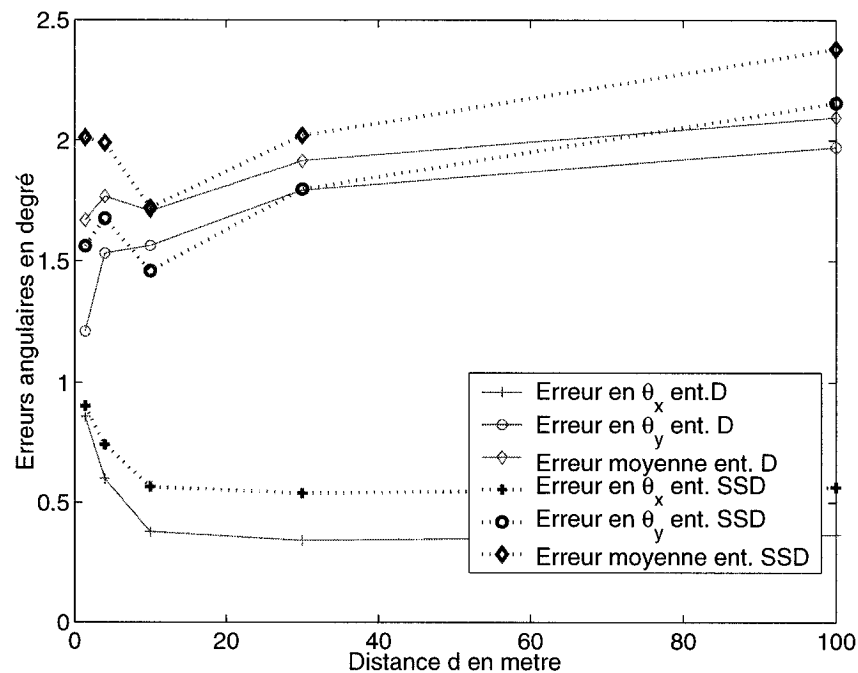


FIG. 4.19 Comparaison des performances de l'entraînement D et de l'entraînement SSD, en fonction de la distance d

Ces deux entraînements donnent sensiblement les mêmes résultats.

La Figure 4.20 est particulièrement intéressante. Elle montre les erreurs angulaires (moyenne, en θ_x et θ_y) d'estimation par l'entraînement D et l'entraînement SSD, quand on augmente l'amplitude du coefficient de réflexion C_0 . Ce dernier n'a aucune existence physique. C'est un coefficient positif constant, indépendant de la fréquence, des caractéristiques électriques du matériau réflecteur et de l'angle d'incidence. On le fait varier entre 0 et 0.6.

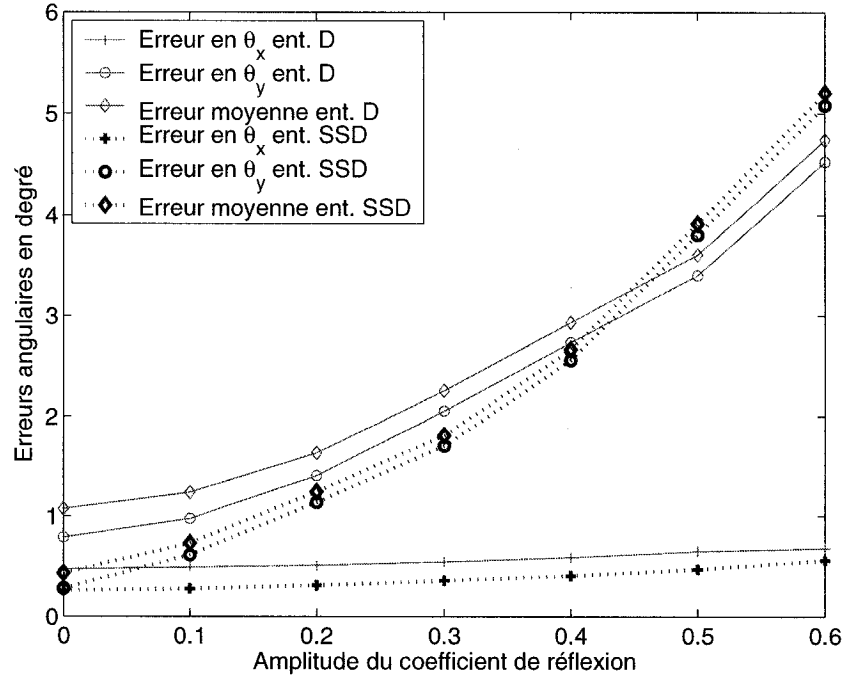


FIG. 4.20 Comparaison des performances de l'entraînement D et de l'entraînement SSD, en fonction de C_0

On voit que l'entraînement D perd de son efficacité lorsque C_0 devient trop important : $\epsilon_{av,y,D} > \epsilon_{av,y,SSD}$ pour $C_0 > 0.43$. Par contre, si la réflexion est de faible amplitude, l'entraînement D donnera de meilleurs résultats. On pourrait réduire encore les erreurs angulaires d'estimation testées pour des réflexions d'amplitude élevée avec l'entraînement SSD en changeant les points d'entraînement. Il suffirait d'augmenter la taille de la matrice $(\Gamma_v)_{k_1,k_2}$, en prenant en plus des coefficients Γ_v compris entre 0.6 et 1.

Conclusion : Le choix de l'entraînement le plus adapté se fera en fonction de l'application que l'on réserve à notre système de DOA. S'il évoluera dans un environnement fortement perturbé par une réflexion, on aura sans doute intérêt à utiliser un entraînement similaire à l'entraînement SSD. Par contre, si la réflexion est faible, l'entraînement D, robuste par essence, remplira pleinement sa fonction.

Pour une application de poursuite automobile évoquée tout au long de ce mémoire, l'entraînement D semble être le plus adéquat pour MLPx. Par contre, pour MLPy, on a besoin de tester nos structures pré-entraînées (selon les entraînements D et SSD) avec des données de validation réelles réalisées avec un plan réflecteur en asphalte. On s'attend en effet à ce que les perturbations induites réellement par un plan en asphalte soient plus ou moins différentes de celles prévues par notre modèle simplifié de réflexion sur route. On pourra alors choisir le meilleur entraînement dans ce cas particulier.

CHAPITRE 5

VALIDATIONS EXPÉRIMENTALES

Pour obtenir des données expérimentales, on a besoin de fabriquer le réseau d'antennes en premier lieu. Comme c'est l'estimation de l'angle θ_y qui est problématique, on va se concentrer sur un réseau de trois antennes selon l'axe Y . Afin de simplifier la conception du système, on évitera de construire le combineur de signaux RF. A la place, on se contentera de mesurer l'amplitude et la phase en sortie de chacune des trois antennes. Ces données seront ensuite utilisées pour générer des points de tests de nos RNAs pré-entraînés. On combinera donc ces signaux pour obtenir les signaux de puissance formant les vecteurs de test. Il est important de noter que seules les antennes ont été fabriquées. Le reste, que ce soit les combineurs de signaux ou les MLPs, sont toujours simulés.

5.1 Synthèse sous HFSS d'un système à trois antennes en guide d'onde

HFSS est un outil de simulation pour le design de structures hautes fréquences, basé sur la méthode des éléments finis. La version du logiciel utilisé est celle d'Ansoft : HFSS 9.0.

5.1.1 Choix et configuration des guides d'onde

Le système a été fabriqué pour fonctionner à 77 GHz. La longueur d'onde λ vaut 3.896 mm à cette fréquence. On a décidé d'utiliser des antennes sous la forme de guides d'onde tronqués, pour leur simplicité à mettre en œuvre. On a choisi des

guides d'onde WR-10, dont la fréquence de coupure (basse) est de 75 GHz, et inférieure à la fréquence d'utilisation de 77 GHz : la longueur a de la section du guide vaut 100 mil (soit 2.54 mm), sa largeur b vaut 50 mil (soit 1.27 mm). On a défini a et b sur la Figure 5.1.

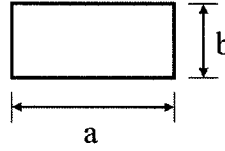


FIG. 5.1 *Définition des longueurs a et b sur une section de guide d'onde*

Traduisons maintenant l'espacement en longueurs d'onde entre les 3 antennes en terme de distance absolue :

$$d_1 = 0.58\lambda = 2.26 \text{ mm} \quad (5.1a)$$

$$d_2 = 1.37\lambda = 5.34 \text{ mm} \quad (5.1b)$$

$$d_2 - d_1 = 0.79\lambda = 3.08 \text{ mm} \quad (5.1c)$$

La distance centre à centre séparant l'antenne 1 de l'antenne 2 vaut 2.26 mm tandis que celle séparant l'antenne 2 de l'antenne 3 vaut 3.08 mm. On peut aligner les guides d'onde selon deux orientations différentes, selon que la polarisation soit plan E ou plan H pour le guide d'onde. Il faut garder en tête que la polarisation est verticale dans tous les cas (polarisation de l'émetteur). La Figure 5.2 représente les deux situations correspondant aux polarisations verticale et horizontale.

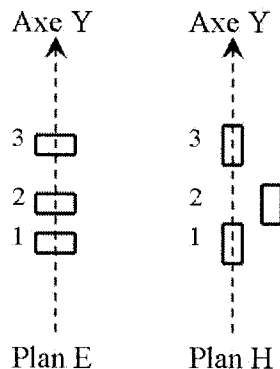


FIG. 5.2 Deux types de configuration possibles pour les guides d'onde : plan E ou plan H

La distance d_1 étant inférieure à la longueur a de la section du guide utilisé, la configuration plan H n'est possible qu'à la condition que l'antenne 2 soit décalée par rapport à l'antenne 1. Ce décalage aura d'autant plus d'influence que l'antenne 2 est excentrée de l'axe Y . En effet, les signaux de sortie des antennes d'axe Y deviennent fonction de θ_y , mais aussi dans ce cas de θ_x , ce que l'on veut éviter par dessus tout. On retiendra donc la configuration plan E pour la fabrication des guides d'onde.

5.1.2 Tracé des guides d'onde

Après avoir choisi le type de guide d'onde, on doit modéliser leur tracé pour que le technicien puisse fabriquer la pièce. On part d'un bloc d'aluminium carré de 3 pouces de côté. On ne peut pas prendre trois guides d'onde rectilignes. En effet, pour pouvoir fixer l'appareil de mesure pour une série de tests ultérieurs, on a besoin d'écarter les guides d'onde entre eux à la sortie, d'un espacement supérieur à

0.5 pouces. On va donc devoir courber d'une certaine manière nos guides d'onde, deux sur trois en fait. On gardera celui du milieu rectiligne (voir Figure 5.3). En courbant les guides d'onde, on rajoute un déphasage supplémentaire des guides 1 et 3 par rapport au guide 2, déphasage qui n'était pas prévu dans nos simulations. Pour éviter d'avoir à déterminer deux inconnues de déphasage, nous avons fait un tracé symétrique (en miroir) des guides 1 et 3. De cette façon, le déphasage entre les guides 1 et 2 est le même qu'entre les guides 2 et 3. On expliquera dans la suite comment on a ensuite déterminé cette inconnue de déphasage. La Figure 5.3 représente le calque HFSS fourni au technicien.

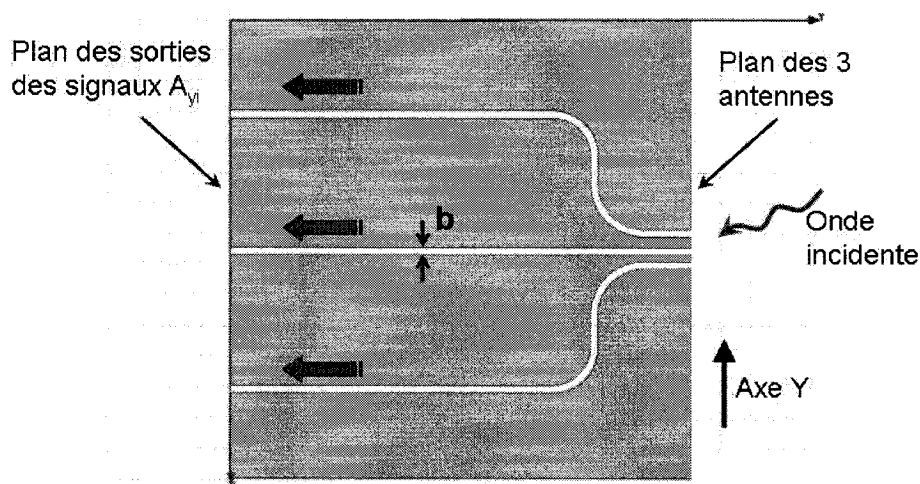


FIG. 5.3 *Tracé des guides d'onde du réseau d'antenne d'axe Y*

Ce tracé a été fait sans tenir compte de considérations de gain ou d'étude des paramètres S . On a supposé qu'un simple guide d'onde serait suffisant pour recevoir les signaux avec un niveau de puissance acceptable.

5.1.3 Calcul de la taille du cornet

Pour avoir plus de signal dans le cas où on n'en recevrait pas assez en sortie des antennes, on a décidé de fabriquer un cornet sectoriel plan H amovible. Généralement, on fixe un cornet en sortie d'un guide d'onde pour réduire le coefficient de réflexion à l'entrée. Pour un cornet sectoriel plan H , la largeur de son ouverture vaut b , i.e. la largeur de la section du guide. Dans notre cas, elle est bien plus grande : elle est égale à la hauteur de la pièce (voir Figure 5.5). De cette façon, on utilise un seul cornet pour le réseau de trois antennes.

Sa conception s'est faite en s'aidant de (Stutzman, 1998). On désigne par A la longueur de l'ouverture du cornet et par R_1 la longueur de la hauteur du triangle défini par le cornet, comme représenté sur la Figure 5.4.

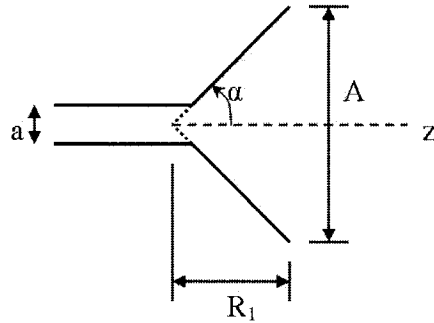


FIG. 5.4 Coupe d'un guide et du cornet plan H dans le plan (X, Z)

On utilise le paramètre t défini p. 303 de (Stutzman, 1998) dont l'expression est donnée par l'équation 5.2 :

$$t = \frac{A^2}{8\lambda R_1} \quad (5.2)$$

Pour une longueur R_1 fixée, la directivité du cornet est optimale pour $t = \frac{3}{8}$. Pour

la longueur d'onde considérée, on arrive à une première équation numérique liant A et R_1 :

$$\frac{A^2}{R_1} = 1.169 \text{ e-} 2 \quad (5.3)$$

On se fixe ensuite une distance arbitraire pour R_1 , donnée dans l'équation 5.4 :

$$R_1 = 10\lambda = 3.90 \text{ cm} \quad (5.4)$$

On en déduit A :

$$A = 6.75 \text{ cm} \quad (5.5)$$

Il est aussi utile de connaître l'angle α d'ouverture du cornet pour sa conception future :

$$\alpha = 40.9^\circ \quad (5.6)$$

5.2 Description du dispositif expérimental

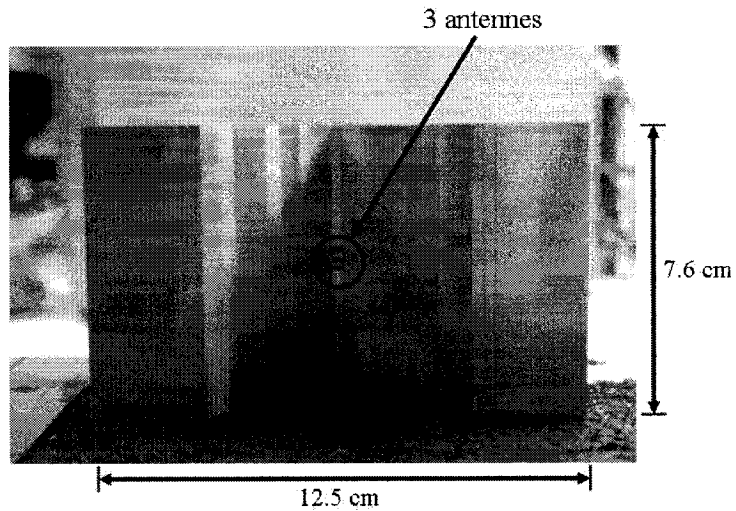


FIG. 5.5 *Photo des trois entrées des antennes en guide d'onde*

Le réseau d'antennes a ensuite été fabriqué dans un bloc d'aluminium. La Figure 5.5 est une photo de notre système, équipé de son cornet sectoriel plan H.

5.2.1 Diagramme de rayonnement des antennes

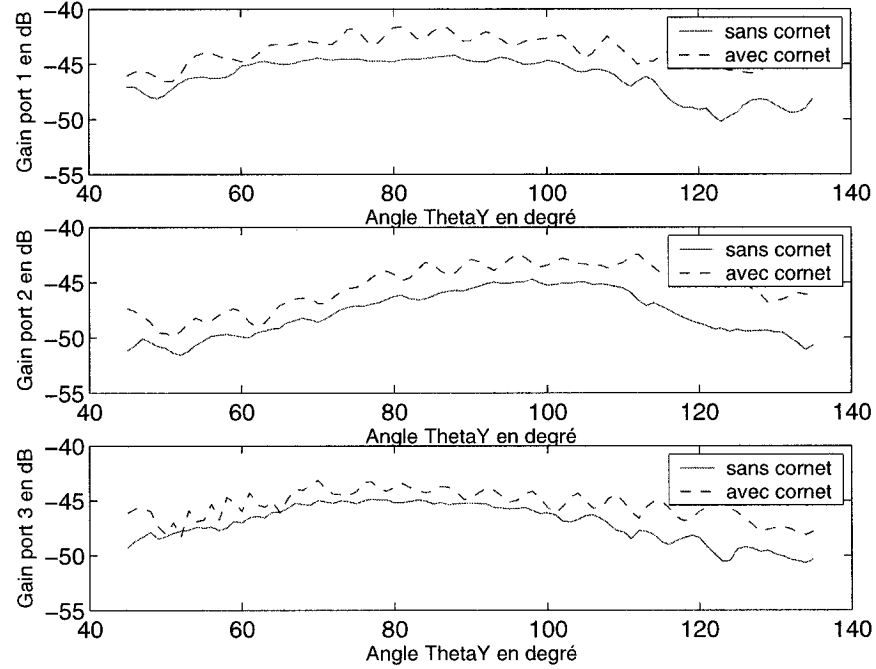


FIG. 5.6 *Diagramme de rayonnement des trois antennes, avec et sans cornet sectoriel plan H*

On a d'abord fait des mesures de diagramme de rayonnement des trois antennes pour vérifier que de simples guides d'onde tronqués peuvent bien recevoir assez de signal pour rendre possible le travail d'estimation (Figure 5.6). Pour cela, on a placé le système dans le plan $\theta_x = 90^\circ$ dans la chambre anéchoïde et on a fait varier l'angle θ_y , i.e. l'élévation. On a fait attention à ce que l'onde plane émise soit bien polarisée verticalement. Cette polarisation restera valable pour toutes les mesures

effectuées par la suite.

On remarque que le cornet sectoriel plan H permet de gagner 3 dB en moyenne pour nos trois antennes. Le balayage a été fait selon l'angle θ_y . On ne voit donc pas l'augmentation de directivité engendrée par le cornet. Elle se verrait pour un balayage selon l'angle θ_x . L'emploi des cornets rend moins régulière cependant le diagramme de rayonnement des antennes. Recevant assez de signal en sortie des guides d'onde, sans cornet, pour effectuer le travail d'estimation, on a préféré faire les mesures sans cornet sectoriel plan H finalement.

5.2.2 Coefficients de réflexion expérimentaux

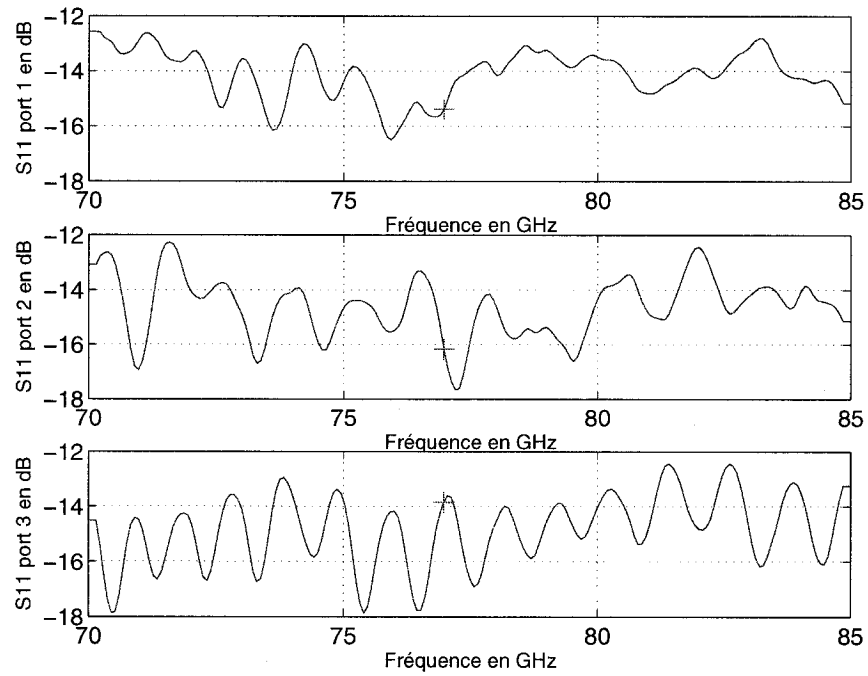


FIG. 5.7 Coefficients de réflexion expérimentaux en entrée de nos trois antennes

Lorsque nous avons fait des premières mesures sans plan réflecteur, il a été possible d'estimer la DOA du signal comme prévu par nos simulations. Cela suggère que l'adaptation à l'entrée de nos guides n'est pas aussi mauvaise que l'on pouvait craindre. Pour s'en convaincre, on a réalisé des mesures de coefficients de réflexion (paramètres $S_{1,1}$) à l'entrée de nos trois antennes autour de 77 GHz à l'aide d'un analyseur de réseaux (Figure 5.7).

Ces coefficients de réflexion sont compris entre -14 et -16 dB environ pour les trois antennes, ce qui est bon compte tenu de la fréquence d'utilisation et du type d'antenne utilisé.

On a également vérifié en pratique que l'atténuation en sortie des guides était faible en plaçant une plaque métallique à l'autre extrémité du guide (celle où on place le port de mesures). On a vérifié que l'on obtenait bien un coefficient de réflexion proche de l'unité.

Notre réseau d'antennes ainsi construit peut donc être utilisé pour estimer des DOAs.

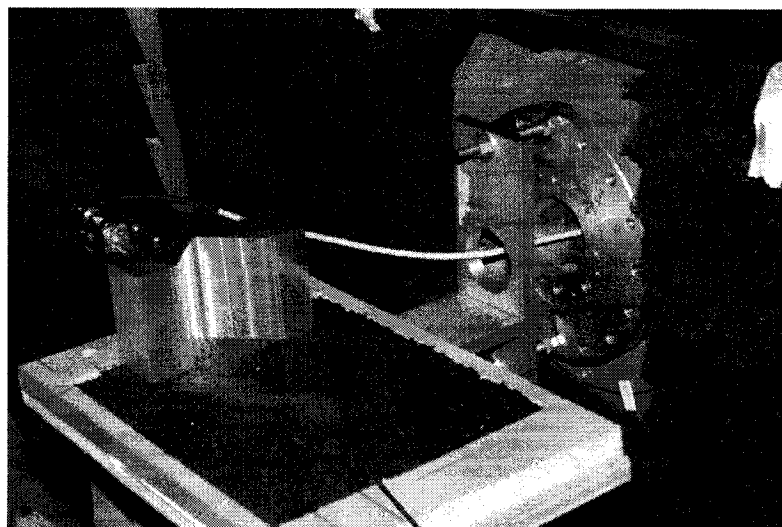
5.2.3 Plans réflecteurs utilisés

Des tests en chambre anéchoïdes nous ont montré que notre réseau d'axe Y pouvait déterminer l'angle θ_y comme prévu en l'absence de plan réflecteur (voir Figure 5.10). On est donc passé à la prochaine étape, qui consiste à fabriquer le plan réflecteur.

Plusieurs matériaux à disposition ont pu être testés : de l'asphalte, du bitume caoutchouteux, du PVC et du contre-plaqué. La réflectivité de ces matériaux a été décrite précédemment dans la Figure 3.5. On a ainsi utilisé quatre plaques différentes du format d'une feuille de papier A4 composées chacune de ces matériaux.

On y a fixé le réseau d'antennes ensuite. Le tout s'adapte au bras articulé de la chambre anéchoïde de manière à prendre des mesures en élévation. La Figure 5.8 représente le dispositif expérimental (le matériau utilisé est de l'asphalte dans ce cas).

Nous n'avons pas pu faire de très longues plaques de matériaux pour que l'on puisse faire des mesures jusque $\theta_y = 45^\circ$ dans le plan $\theta_x = 90^\circ$. En contre-partie, aucune réflexion n'est générée à proprement parler pour des incidentes rasantes, i.e. pour des angles θ_y proches de 90° . Mais des phénomènes de diffraction devraient se produire à cause des bords de la plaque, phénomène totalement mis de côté durant les simulations. Nous verrons dans quelle mesure ils perturberont nos résultats.



Plaque d'asphalte

FIG. 5.8 *Photo du système monté sur le plan réflecteur en asphalte, dans la chambre anéchoïde*

5.3 Résultats expérimentaux sur l'estimation de θ_y dans le plan $\theta_x = 90^\circ$

Nous avons donc procédé à plusieurs séries de mesures pour obtenir les sorties complexes de nos trois antennes. Pour se faire, nous avons fait un balayage en élévation pour des angles θ_y variant de 90° à 45° , dans le plan $\theta_x = 90^\circ$ et avec un des quatre plans réflecteurs. Ne disposant que d'un seul port de mesures, nous avons itéré ce protocole trois fois, pour chacune des antennes, en faisant très attention à ne pas bouger le système par rapport au bâti entre les trois séries de mesures.

Ces données sont ensuite importées dans Matlab[®] sous la forme de fichiers texte. On les combine pour former les huit signaux de puissance. On utilise ainsi des vecteurs de huit composantes pour alimenter MLPy. Il est important de rappeler que MLPy a déjà été pré-entraîné, selon les différentes méthodes vues dans le chapitre précédent. Les entraînements D, DRM et SSD vont ainsi pouvoir être testés et comparés dans des conditions proches de la réalité.

Il reste un problème à résoudre cependant avant de passer à cette étape : celui de la détermination du déphasage additionnel dans les guides 1 et 3 par rapport au guide 2. On a procédé de la façon suivante pour lever cette inconnue. Des mesures ont été faites sans plan réflecteur, sur les trois antennes, pour des angles θ_y variant de 135° à 45° , dans le plan $\theta_x = 90^\circ$. Nous avons ensuite rajouté un déphasage connu δ sous Matlab sur les données de sortie des antennes 1 et 3. Nous avons ensuite testé MLPy entraîné avec l'entraînement D (le plus sûr car le plus simple) et calculé les erreurs d'estimation correspondant à ce déphasage. Plusieurs valeurs ont ainsi pu être testées et de proche en proche (par dichotomie), nous avons constaté qu'un déphasage $\delta = 0.50\pi$ procurait de très bons résultats.

On peut essayer de retrouver cette valeur en déterminant la différence de chemin parcourue par les ondes dans les différents guides. Elle s'obtient géométriquement

à l'aide de la Figure 5.3. Connaissant les centres et rayons de courbure des arcs de cercle constituant les guides 1 et 3, il est possible de déterminer la longueur géométrique des guides 1 à 3 et par suite leur différence de longueur. Pour effectuer ce calcul, on considère le trajet situé au milieu des guides. Soit Δ_l la différence de parcours entre le guide 2 d'une part, et les guides 1 et 3 d'autre part :

$$\Delta_l = 13.2 \text{ mm} \quad (5.7)$$

Soit $\tilde{\delta}$ la valeur de déphasage trouvée géométriquement. Elle s'obtient grâce à l'équation 5.8.

$$\tilde{\delta} = \beta_g \Delta_l \quad (5.8)$$

β_g est la constante de propagation du mode $TE_{1,0}$ (le mode fondamental de nos guides). Elle est donnée par l'équation 5.9.

$$\beta_g = \sqrt{\left(\frac{2\pi}{\lambda}\right)^2 - \left(\frac{\pi}{a}\right)^2} \quad (5.9)$$

On trouve que $\beta_g = 1035 \text{ rad/m}$. On en déduit finalement le déphasage entre nos guides :

$$\tilde{\delta} = 0.36\pi \mod 2\pi \quad (5.10)$$

Cette valeur n'est pas exactement celle trouvée précédemment par dichotomie. Elle donne de moins bons résultats. Cette différence peut provenir du fait que la mesure (via le connecteur) en sortie des guides ne se fait pas exactement dans le même plan en pratique. Dans la suite, on prendra $\delta = 0.50\pi$.

Si on désigne par $A_{yi,\text{réel}}$, $i = 1, 2, 3$ les signaux mesurés en sortie des antennes, on obtient facilement les signaux rephasés $A_{yi,\text{rephasé}}$, $i = 1, 2, 3$ utilisés dans le

combinateur :

$$A_{yi,\text{rephasé}} = A_{yi,\text{réel}} \exp(-0.50\pi j), \quad i = 1, 3 \quad (5.11a)$$

$$A_{y2,\text{rephasé}} = A_{y2,\text{réel}} \quad (5.11b)$$

Les Figures 5.9 et 5.10 permettent de voir l'amélioration spectaculaire quant à la précision de l'estimation de θ_y quand on utilise des signaux rephasés. L'explication de cette différence est simple : si on ne rephase pas les signaux complexes de sortie des antennes, on obtient des signaux de puissance en sortie du combineur qui sont très éloignés de ceux utilisés pour entraîner MLPy. Le travail d'estimation est alors impossible. L'erreur d'estimation de MLPy vaut $\epsilon_{av,y} = 1.70^\circ$ quand celui-ci a été entraîné avec l'entraînement D. Durant le test, le bruit de mesure est inconnu. Nous n'avons pas placé de plan réflecteur dans la chambre anéchoïde, ni rajouté de bruit sous Matlab[®] par la suite. Nous avons bien sûr rephasé les signaux.

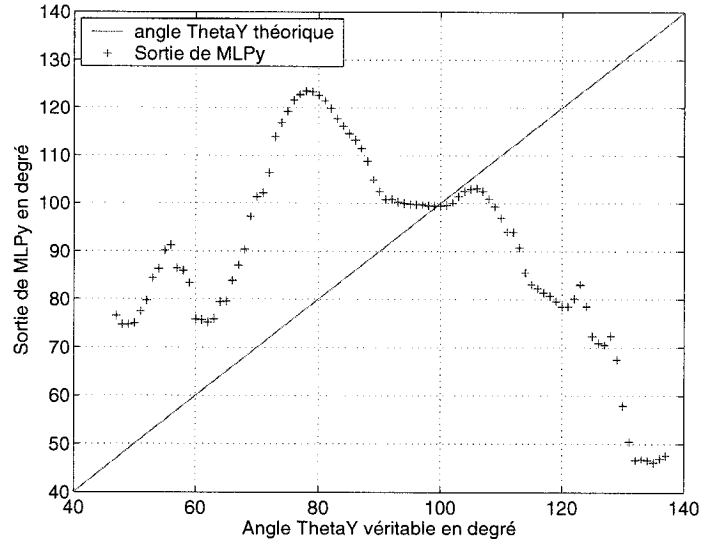


FIG. 5.9 Résultats d'estimation de MLPy (entraînement D) sans plan réflecteur, sans bruit additif, avec signaux $A_{yi,\text{réel}}$

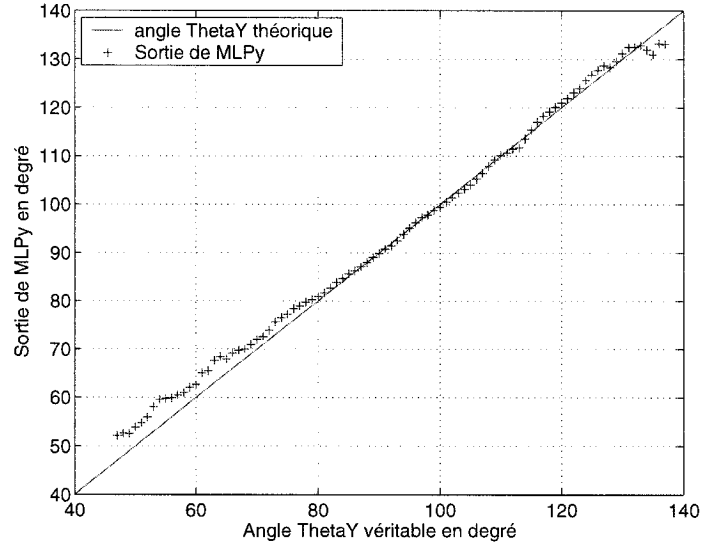


FIG. 5.10 Résultats d'estimation de MLPy (entraînement D) sans plan réflecteur, sans bruit additif, avec signaux $A_{yi, \text{rephasé}}$

La valeur du déphasage étant connue, on l'utilisera pour tester les différentes situations de réflexion (avec différents matériaux) sur MLPy, pré-entraîné selon les différentes méthodes vues dans le chapitre précédent.

5.3.1 Avec entraînement sans réflexion

On va maintenant tester le même RNA que le paragraphe précédent. Cette fois, on a placé un plan réflecteur dans la chambre anéchoïde. Les mesures ne se font plus qu'entre 45 et 90° pour cause d'obstruction du signal pour des angles d'élévation plus élevés. On présente dans cette section les figures d'estimation de l'entraînement D (Figures 5.11, 5.12, 5.13 et 5.14), ainsi qu'un tableau récapitulatif de ses performances (Tableau 5.1).

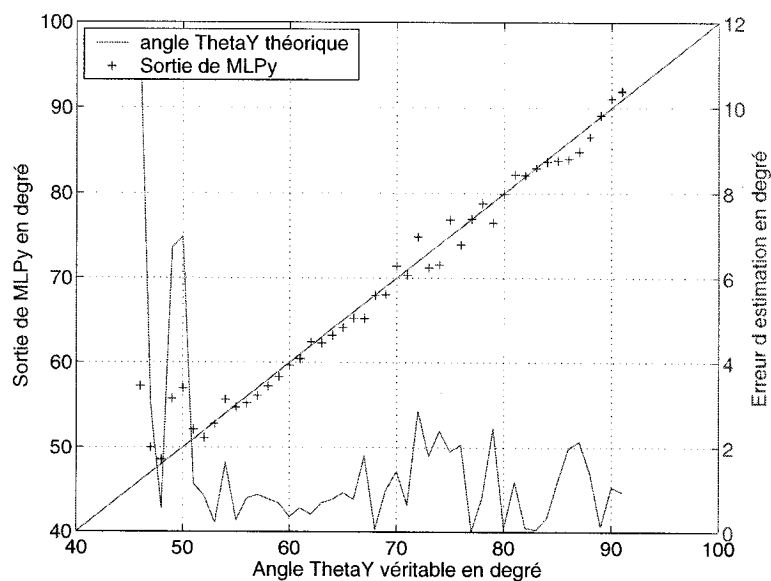


FIG. 5.11 Résultats d'estimation de MLPy (entraînement D) testé avec un plan réflecteur en PVC

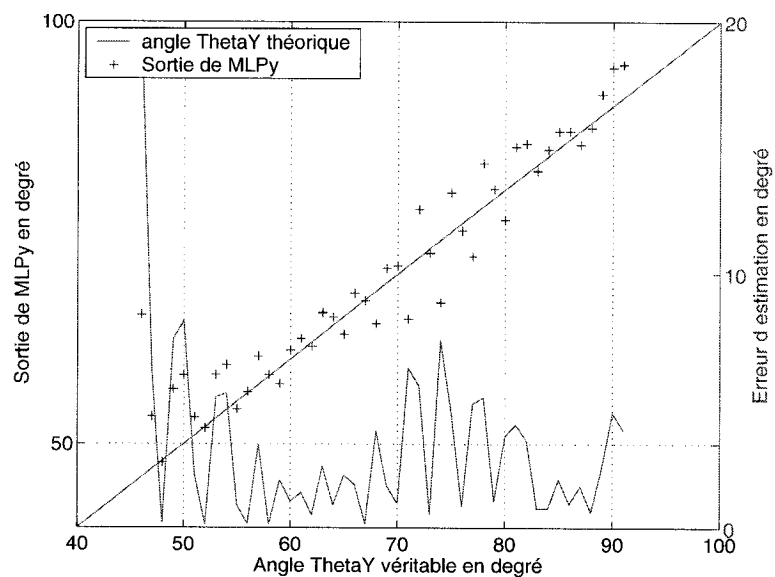


FIG. 5.12 Résultats d'estimation de MLPy (entraînement D) testé avec un plan réflecteur en contre-plaqué

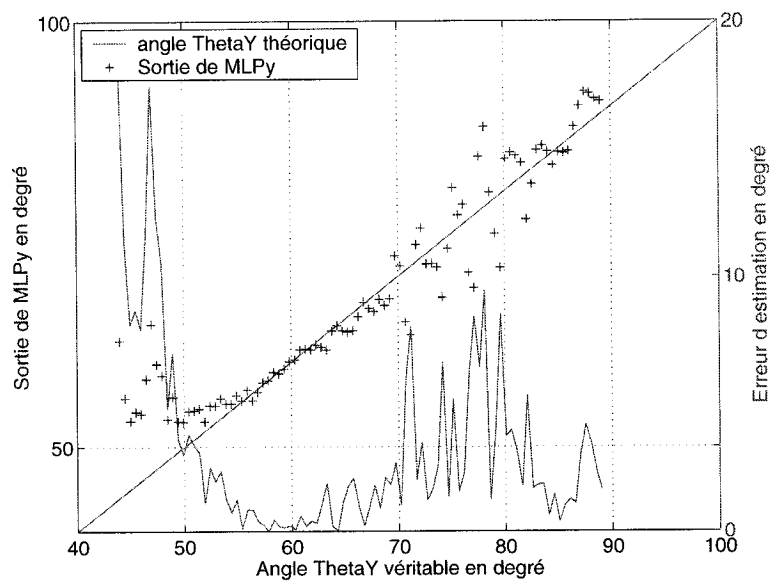


FIG. 5.13 Résultats d'estimation de MLPy (entraînement D) testé avec un plan réflecteur en bitume caoutchouteux

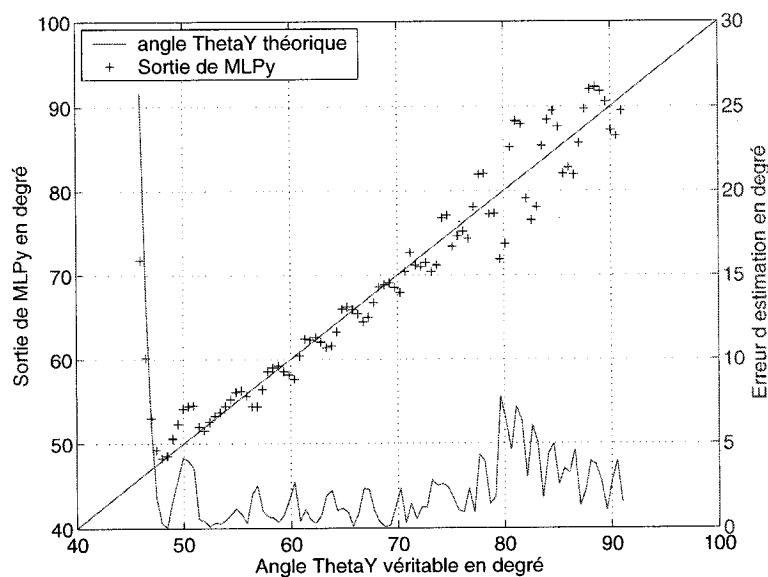


FIG. 5.14 Résultats d'estimation de MLPy (entraînement D) testé avec un plan réflecteur en asphalte

	$\epsilon_{av,y}$ avec ent. D
avec plan réflecteur en PVC	1.52°
avec plan réflecteur en bois	3.09°
avec plan réflecteur en bitume caoutchouteux	3.29°
avec plan réflecteur en asphalte	2.50°

TAB. 5.1 Performances de l'entraînement D testé avec différents plans réflecteurs

5.3.2 Avec entraînement avec réflexion multi-configuration

On procède de la même façon ici pour tester le réseau MLPy entraîné avec l'entraînement DRM. On donne donc les figures d'estimation de l'entraînement DRM (Figures 5.15, 5.16, 5.17 et 5.18), ainsi que le tableau récapitulatif de ses performances (Tableau 5.2).

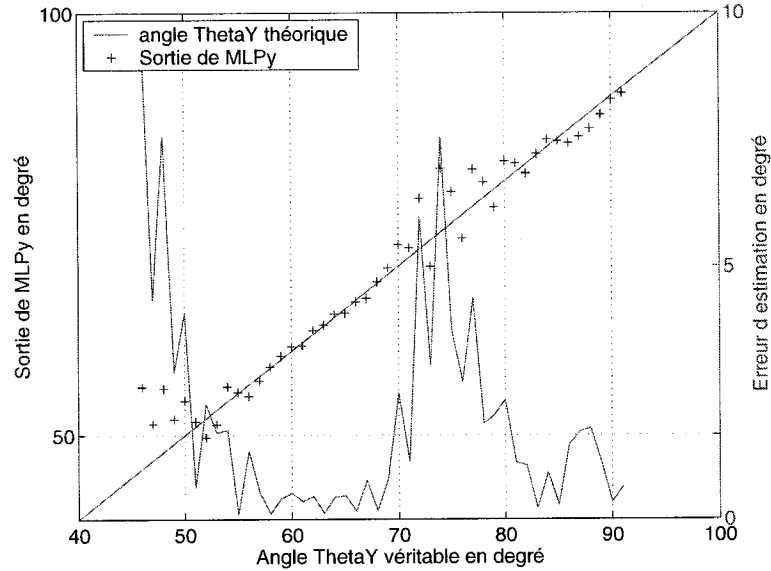


FIG. 5.15 Résultats d'estimation de MLPy (entraînement DRM) testé avec un plan réflecteur en PVC

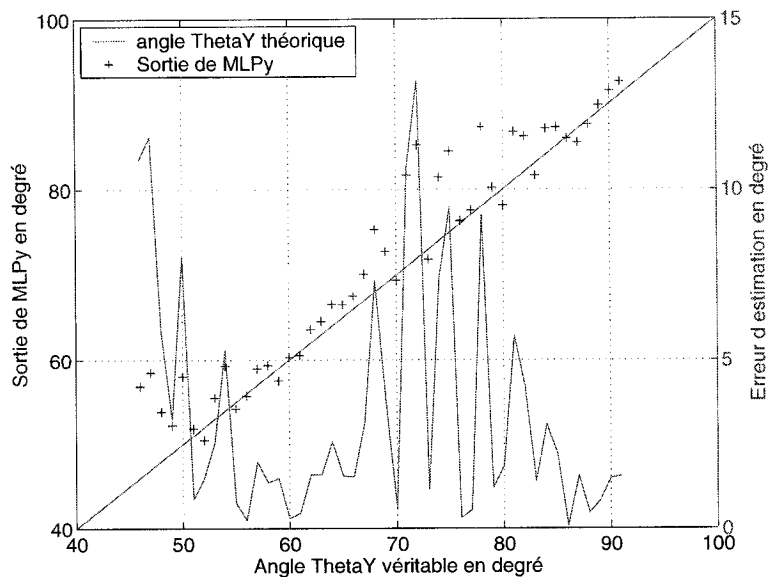


FIG. 5.16 Résultats d'estimation de MLPy (entraînement DRM) testé avec un plan réflecteur en contre-plaqué

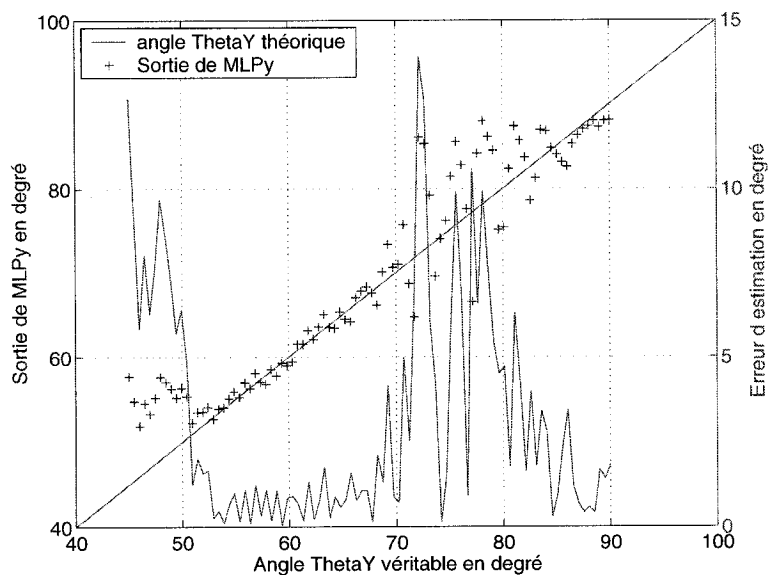


FIG. 5.17 Résultats d'estimation de MLPy (entraînement DRM) testé avec un plan réflecteur en bitume caoutchouteux

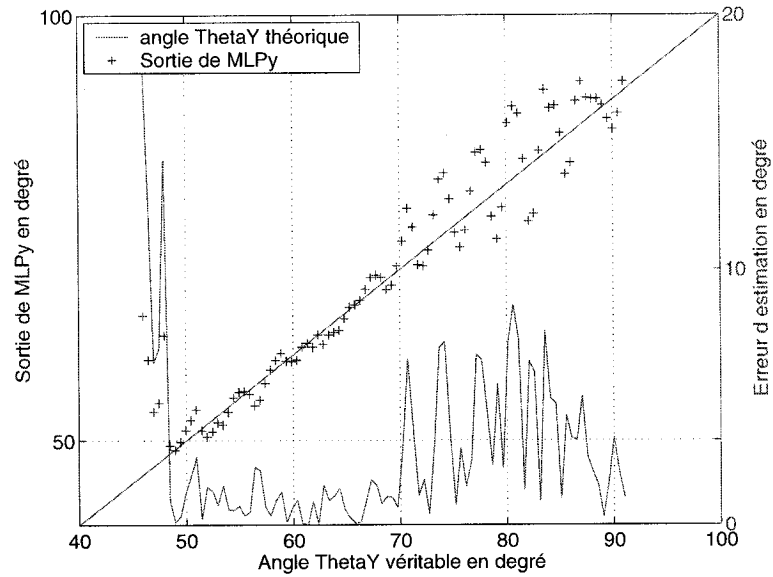


FIG. 5.18 Résultats d'estimation de MLPy (entraînement DRM) testé avec un plan réflecteur en asphalte

	$\epsilon_{av,y}$ avec ent. DRM	gain p/r à ent. D
PVC	1.95°	-0.43°
bois	3.43°	-0.34°
bitume	3.22°	+0.07°
asphalte	2.83°	-0.33°

TAB. 5.2 Performances de l'entraînement DRM testé avec différents plans réflecteurs et comparé avec l'entraînement D

5.3.3 Avec entraînement bi-signal décalé

Enfin, on teste le réseau MLPy entraîné avec l'entraînement SSD. On en donne donc les figures d'estimation (Figures 5.19, 5.20, 5.21 et 5.22), ainsi que le tableau récapitulatif de ses performances (Tableau 5.3).

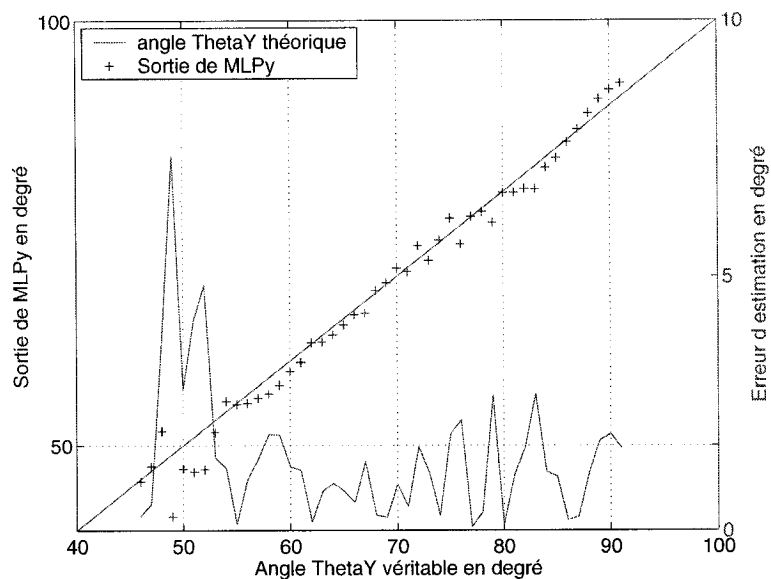


FIG. 5.19 Résultats d'estimation de MLPy (entraînement SSD) testé avec un plan réflecteur en PVC

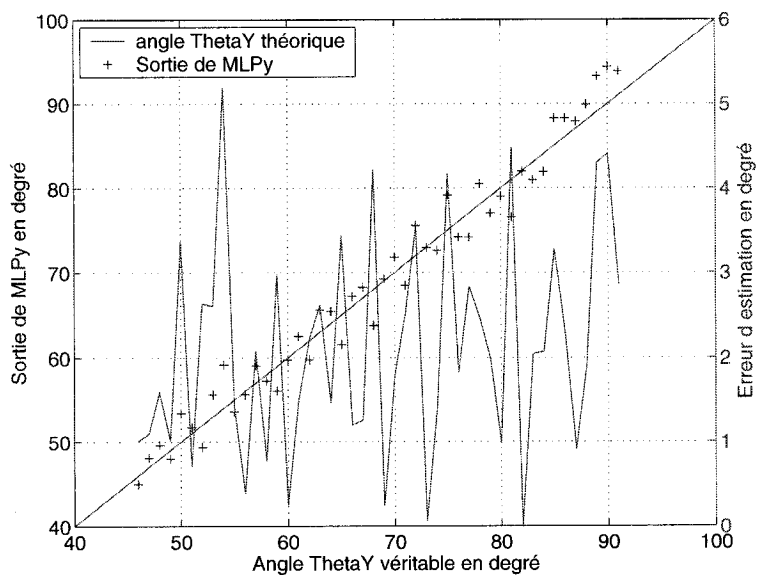


FIG. 5.20 Résultats d'estimation de MLPy (entraînement SSD) testé avec un plan réflecteur en contre-plaqué

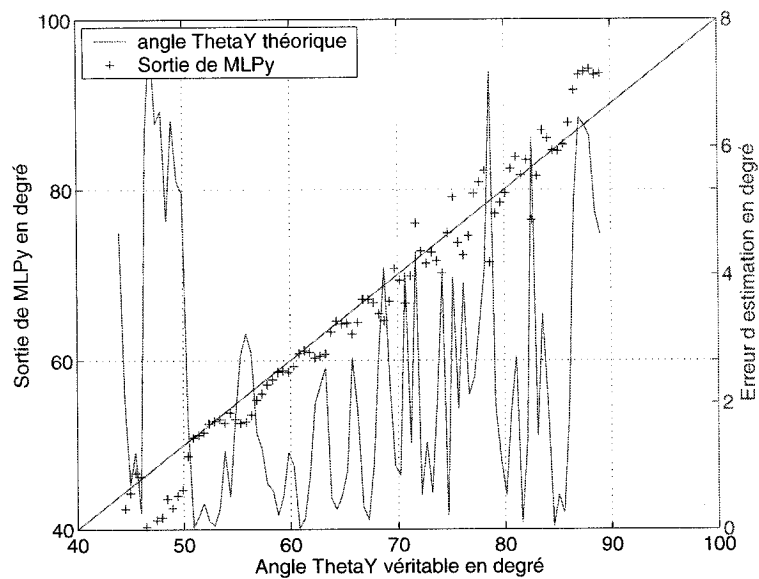


FIG. 5.21 Résultats d'estimation de MLPy (entraînement SSD) testé avec un plan réflecteur en bitume caoutchouteux

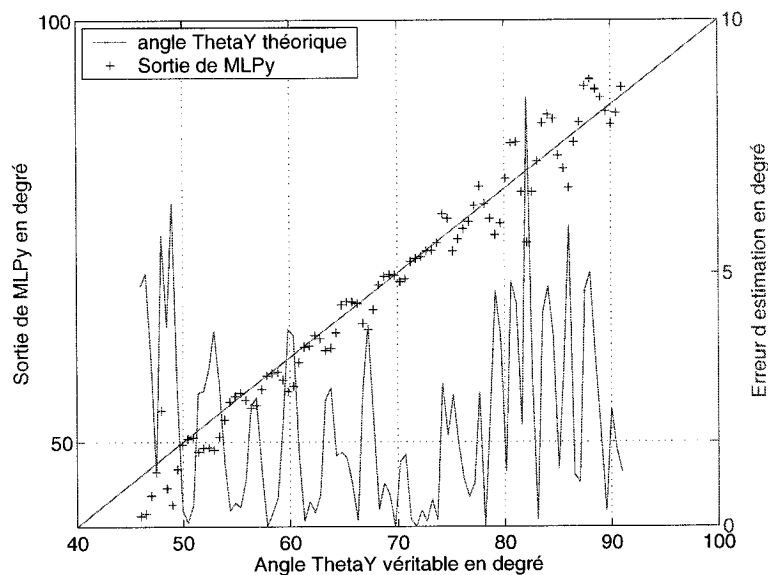


FIG. 5.22 Résultats d'estimation de MLPy (entraînement SSD) testé avec un plan réflecteur en asphalte

	$\epsilon_{av,y}$ avec SSD	gain p/r à ent. D
PVC	1.44°	+0.08°
bois	2.12°	+0.97°
bitume	2.32°	+0.97°
asphalte	2.06°	+0.44°

TAB. 5.3 *Performances de l'entraînement SSD testé avec différents plans réflecteurs et comparé avec l'entraînement D*

5.3.4 Analyse des résultats expérimentaux

On remarque que l'on peut très clairement classer ces types d'entraînement selon leurs performances à estimer θ_y en présence de réflexion : on constate que l'entraînement SSD est meilleur que l'entraînement D, lui-même meilleur que l'entraînement DRM.

On peut quantifier ces performances :

- l'entraînement SSD est en moyenne 21 % plus précis que l'entraînement D sur l'ensemble des matériaux testés ;
- l'entraînement D est en moyenne 11 % plus précis que l'entraînement DRM sur l'ensemble des matériaux testés.

Les erreurs angulaires rencontrées expérimentalement sont en très bon accord avec les résultats en simulation, en ce qui concerne les entraînements SSD et DRM. Les expériences confirment la puissance de l'entraînement SSD : on a eu raison d'entraîner un RNA avec une réflexion, indépendante du signal incident ou du matériau réflecteur. Ceci est surtout vrai en pratique, où la rugosité et l'hétérogénéité du matériau réflecteur font que l'on s'éloigne de notre modèle de réflexion sur route, à la base de notre entraînement DRM.

L'entraînement D est moins performant que prévu par les simulations. Ceci s'explique peut-être par le fait que la réflexion parasite ne se modélise pas aussi simplement qu'en simulation. Le point positif au niveau de la corroboration des résultats simulés et expérimentaux avec l'entraînement D réside dans le fait qu'il est d'autant moins bon par rapport à l'entraînement SSD, que la réflexion est forte, i.e. que la réflectivité du matériau est grande. Dans notre cas, c'est le contre-plaqué qui est le matériau le plus réfléchissant. L'amélioration relative atteint alors 32 %.

On s'aperçoit que l'erreur d'estimation trouvée expérimentalement pour des angles proches de 90° (élévation principale) est plus grande que celle prévue en simulations pour les mêmes angles. Ceci est sans doute dû aux effets de diffraction introduits par les bords du matériau réflecteur (il n'est plus infini comme lors des simulations). Ils se produisent quand le faisceau incident rencontre le bord de la dalle réflectrice, i.e. pour des élévations proches de 90° justement.

5.4 Résultats expérimentaux sur l'estimation de θ_y sur une portion d'espace

Dans cette section, on va faire un balayage en élévation et en azimuth. θ_y varie entre 45° et 90° , θ_x varie entre 80° et 135° par intervalle de 1° cette fois. Un bloc absorbant dans la chambre fait que l'on ne peut pas prendre des mesures d'azimut inférieures à 80° . De plus, on a testé uniquement les points situés dans le cône d'intérêt. Ceci explique pourquoi nous avons moins de points de tests pour des angles θ_y proches de 45° .

5.4.1 Avec l'entraînement sans réflexion

On a représenté comme précédemment les estimations de MLPy, ainsi que ses erreurs angulaires, sur la Figure 5.23.

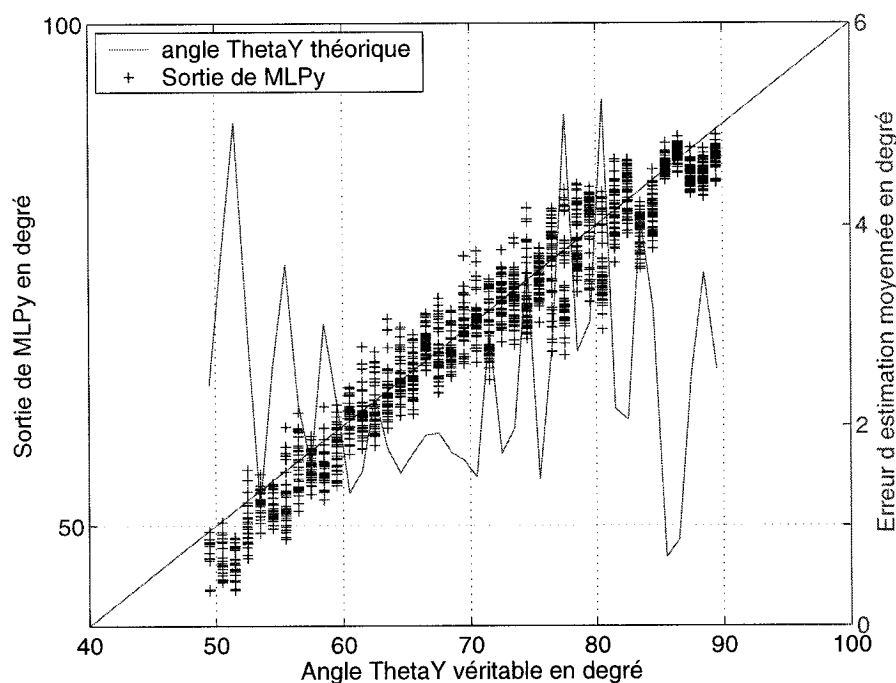


FIG. 5.23 Résultats d'estimation de MLPy testé avec un plan réflecteur en asphalte, avec l'entraînement D (mesures en azimuth et élévation)

Pour une élévation donnée, on remarque que l'on a plusieurs points, correspondant aux différentes valeurs de θ_x ($80^\circ < \theta_x < 135^\circ$). La courbe représentant l'erreur d'estimation faite sur θ_y est réalisée en prenant la moyenne sur les différentes erreurs (en valeur absolue) effectuées pour chaque azimuth. Il est intéressant de voir comment se répartissent ces erreurs en fonction de l'élévation, mais de l'azimut également. La Figure 5.24 représente l'erreur en fonction de ces deux variables.

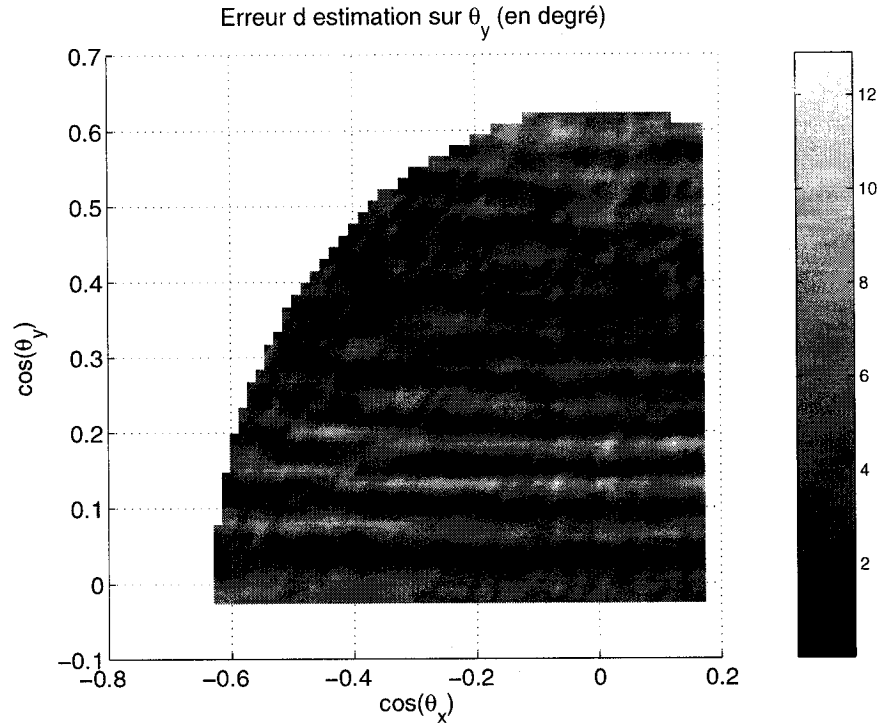


FIG. 5.24 Résultats d'estimation en 3D de MLPy testé avec un plan réflecteur en asphalte, avec l'entraînement D (mesures en azimut et élévation)

On remarque que l'erreur $\epsilon_{av,y}$ est avant tout fonction de θ_y plutôt que de θ_x (voir les franges horizontales sur la Figure 5.24). C'est en accord avec nos simulations puisque les signaux normalisés $X_{iy,n}$ (et $X_{iy,r,n}$) en sortie du combineur sont fonctions de θ_y principalement. On a des erreurs d'estimation importantes quand les signaux $X_{iy,r,n}$ sont assez différents des signaux $X_{iy,n}$ (obtenus sans réflexion). Les interférences de la réflexion avec le signal incident au niveau des signaux de puissance, se répercutent en terme d'erreur angulaire sur l'estimation de θ_y .

5.4.2 Avec l'entraînement bi-signal décalé

On fait de même avec l'entraînement SSD cette fois. Les Figures 5.25 et 5.26 représentent les erreurs d'estimation de θ_y en fonction de l'élévation pour la première, et en fonction de l'azimut et de l'élévation pour la seconde.

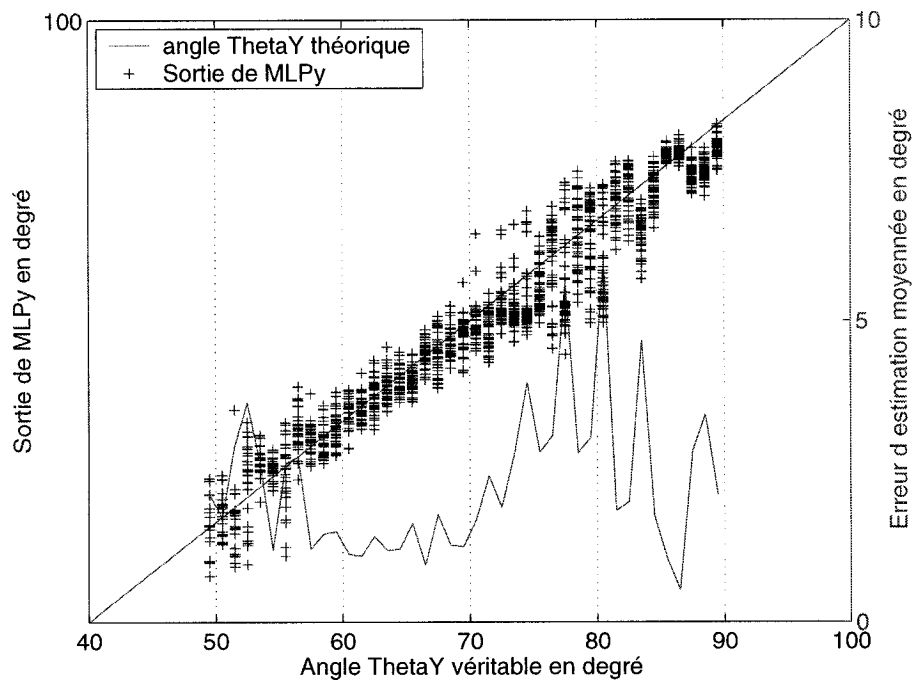


FIG. 5.25 Résultats d'estimation de MLPy testé avec un plan réflecteur en asphalte, avec l'entraînement SSD (mesures en azimut et élévation)

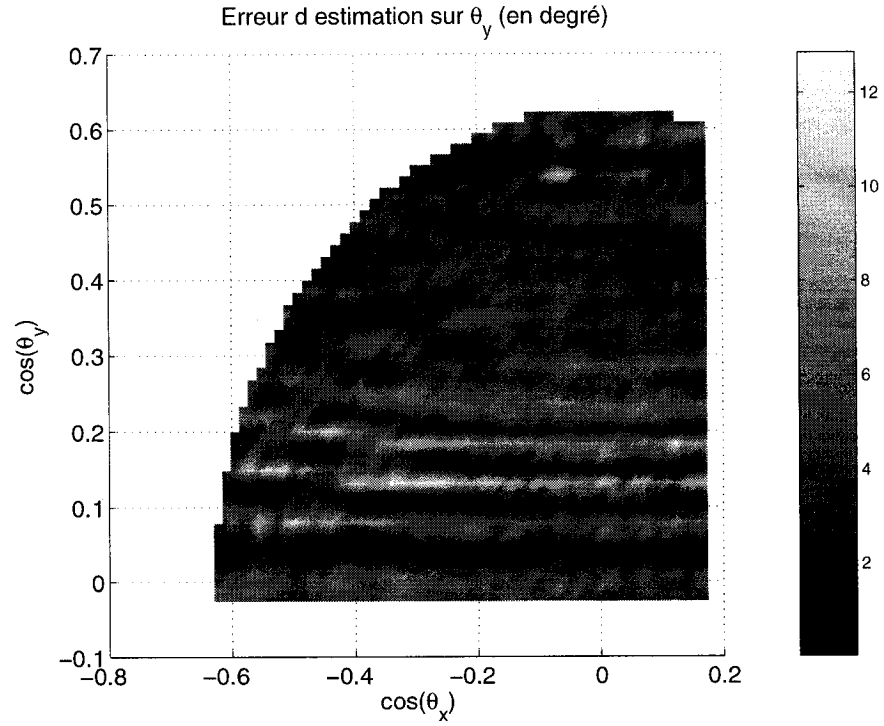


FIG. 5.26 Résultats d'estimation en 3D de MLPy testé avec un plan réflecteur en asphalte, avec l'entraînement SSD (mesures en azimuth et élévation)

5.4.3 Analyse des résultats

Il est difficile de trancher entre les deux types d'entraînement. Ils semblent donner des résultats équivalents, ce que confirme le Tableau 5.4 :

	$\epsilon_{av,y}$
ent. D	2.47°
ent. SSD	2.33°

TAB. 5.4 Performances des entraînements D et SSD, testés avec un plan réflecteur en asphalte (mesures en azimuth et élévation)

L'erreur d'estimation de θ_y est la même avec l'entraînement D selon que l'on fasse des mesures dans un plan ou dans l'espace (dans le cône d'intérêt toujours bien sûr). Par contre, on a une légère augmentation de l'erreur avec l'entraînement SSD quand on fait des mesures en azimut et élévation, l'entraînement SSD restant meilleur que l'entraînement D. On peut en effet remarquer sur la Figure 5.26 que les erreurs sont plus grandes (pour une élévation donnée) quand on s'éloigne de l'azimut centrale ($\theta_x = 90^\circ$). Ceci explique cette diminution de précision.

Il serait intéressant de faire d'autres tests, avec d'autres matériaux réflecteurs, plus réfléchissants surtout. C'est le cas lorsque la route est mouillée. La constante diélectrique de l'eau étant beaucoup plus élevée que l'asphalte ou même la terre ($\epsilon_r = 80$ pour l'eau), les effets de la réflexion s'en trouvent accrus. On pourrait alors de nouveau comparer les entraînements D et SSD, et trancher plus catégoriquement en faveur de l'une des deux méthodes. Avec les résultats à disposition, l'entraînement SSD apparaît cependant comme le plus performant.

CONCLUSION

Nous avons exposé dans ce mémoire les différentes améliorations que nous avons apportées au système de DOA élaboré par Michael Coudyser et Nelson Fonseca, afin de le rendre robuste à une réflexion parasite.

Comme point de départ de notre étude, nous avions un système de DOA simple et peu coûteux composé de deux réseaux d'antennes, de deux combineurs de signaux, d'un convertisseur A/N, d'un filtre (opération de moyennage pour réduire le bruit) et de trois réseaux de neurones de type MLP. Nous avons optimisé les divers composants de ce système pour réduire l'erreur angulaire d'estimation de la DOA.

Ayant en vue une possible application de poursuite automobile pour notre système, la première étape a consisté à établir un modèle simple mais néanmoins représentatif de la réalité physique, d'une réflexion sur une route. On a ainsi pu introduire cette perturbation dans le travail d'estimation. De cette manière, on a pu tester différentes configurations et choisir la plus adaptée :

- on s'est aperçu qu'il était préférable d'utiliser des antennes à polarisation verticale plutôt que des antennes à polarisation circulaire ou pire, horizontale ;
- on a vérifié qu'une paire de réseaux de trois antennes donnait de meilleurs résultats qu'une paire de réseaux de deux antennes ;
- on a vu l'intérêt de découpler l'estimation de θ_x et θ_y par l'intermédiaire de deux RNAs différents, MLPx et MLPy, l'erreur angulaire moyenne étant bien plus élevée pour θ_y que pour θ_x ;
- on a surtout testé différents types d'entraînement pour MLPx et MLPy : on a vu dans quelles conditions il était préférable d'utiliser un entraînement simple, avec un seul signal (entraînement sans réflexion) et quand il était plus judicieux

d'avoir recours à un entraînement plus élaboré, prenant en compte la donnée de deux signaux, le principal et la réflexion (entraînement bi-signal décalé).

Nos simulations sous Matlab[®] ont été corroborées par des données réelles provenant de la mesure en amplitude et en phase de la sortie des trois antennes de l'axe Y quand on plaçait le réseau au dessus d'un plan réflecteur au sein d'une chambre anéchoïde. Les résultats de simulation et expérimentaux sont en bon accord : l'erreur angulaire moyenne d'estimation de θ_y est d'environ 2° dans les deux cas pour des signaux contenus dans le cône d'intérêt, d'ouverture 90° .

Un point important à signaler réside dans le fait que nos réseaux de neurones sont toujours entraînés indépendamment des diagrammes de rayonnements des antennes utilisées. Le gain par défaut est en $\cos \theta$ durant la phase d'entraînement. Une fois entraînés, nos RNAs sont capables de retrouver la DOA des signaux qu'ils reçoivent en pratique et ce même si les diagrammes de rayonnement réels des antennes ne sont pas en $\cos \theta$. Cela signifie que notre système serait fonctionnel immédiatement, sans phase d'étalonnage complexe.

Recommandations pour travaux futurs :

Il faut garder à l'esprit que notre système de DOA est continu, i.e. qu'il fournit un flux ininterrompu d'estimations angulaires pour peu que le signal reste dans le cône d'intérêt. Or, tout notre travail s'est porté sur une estimation de la DOA à un instant précis, sans tenir compte des estimations passées, un peu comme si on ne s'intéressait qu'à une seule image d'un film. On pourrait très certainement améliorer la réponse de notre système en tenant compte des estimations précédentes.

Il serait intéressant de poursuivre cette étude au niveau RF en fabriquant les

autres éléments constitutifs du système, en particulier le combineur de signaux et le récepteur RF. En effet, divers phénomènes difficiles à modéliser ont lieu dans ces deux structures, perturbant d'autant la qualité de l'estimation. On pourrait alors avoir une bonne approximation des performances réelles de notre système, en présence de bruit et d'une réflexion parasite.

RÉFÉRENCES

Charpentier, E. ; Laurin J.J. (1999). *An implementation of a direction-finding antenna for mmobile communications using a neural network*, IEEE transactions on antenna and propagation, vol.47, juillet 1999.

Chen, Y. H. ; Chiang C. T. (1994). *Kalman-based estimator for DOA estimation*, IEEE transactions on signal processing, vol.42, décembre 1994.

Coudyser, M. (2003). *Repérage de la direction d'arrivée d'un faisceau par rapport à un réseau d'antennes disposées sur deux axes et à l'aide d'un réseau de neurones*, Thèse M.Sc.A, Ecole Polytechnique de Montréal.

El Zooghby, A. H. ; Christodoulou, C.G. ; Georgiopoulos M. (1997). *Performance of radial-basis function networks for direction of arrival estimation with antenna arrays*, IEEE transactions on antennas and propagation, vol.45, novembre 1997.

El Zooghby, A. H. ; Christodoulou, C.G. ; Georgiopoulos M. (1997). *A neural network-based smart antenna for multiple source tracking*, IEEE transactions on antennas and propagation, vol.48, mai 2000.

Feder, M. ; Weistein E. (1988). *Parameter estimation of superimposed signals using the EM algorithm*, IEEE transactions on signal processing, vol.36, avril 1988.

Fonseca, N. (2003). *Etude, Conception et modélisation d'antennes à polarisation circulaire - Application à un système de poursuite pour télécommunications*

par satellite basse orbite, Thèse M.Sc.A, Ecole Polytechnique de Montréal.

Haykin, S. (1999). *Neural Networks : a comprehensive foundation, second edition*, Prentice Hall, 1999.

Jordan, E. C. (1950). *Electromagnetic waves and radiating systems*, Prentice Hall, 1950.

Lo, T. ; Leung H. ; Litva J. (1994). *Radial basis function neural network for direction-of-arrivals estimation*, IEEE signal processing letters, vol.1, février 1994.

Lo, T. K. Y. ; Leung H. ; Litva J. (1994). *Artificial neural network for aoa estimation in a multipath environment over the sea*, IEEE journal of oceanic engineering, vol.19, octobre 1994.

Ma, C. W. ; Teng C. C. (1999). *Fuzzy neural network approach for 2-D direction finding in multipath environments*, IEE proceedings on radar and sonar navigation, vol.146, avril 1999.

Roy, R.H. ; Kailath T. (1989). *ESPRIT - Estimation of signal parameters via rotational invariance techniques*, IEEE transactions on acoustics, vol.37, juillet 1989.

Schmidt, R.O. (1986). *Multiple emitter location and signal parameter estimation*, IEEE transactions on antennas and propagation, AP-34, Mars 1986.

Shieh, C. S.; Lin C. T. (2000). *Direction of arrival estimation based on phase differences using neural fuzzy network*, IEEE transactions on antenna and propagation, vol.48, juillet 2000.

Stutzman, W. L.; Thiele G. A. (1998). *Antenna theory and design*, Wiley, 1998.

Wong, T.; Lo, T.; Leung H.; Litva J.; Bosse E. (1993). *Low-angle radar tracking using radial basis function neural network*, IEE proceedings, vol.140, octobre 1993.

Tableau de constantes diélectriques.

www.clippercontrols.com/info/dielectric/constants.html